Processors and Peripherals

QUICK REFERENCE GUIDE

Document No. 60311

May 1982

Reference Guide Conventions

The following conventions are adhered to throughout this guide.

1. All numbers not subscripted are assumed to be decimal unless noted in a table or chart.
2. A "$" appearing anywhere in this text other than character charts, shall denote an unused or unassigned item.
3. All items of information are generated for those model codes with which they are listed and cannot be assumed to apply to any models not listed.

This guide is designed to aid the programmer and system analyst in using Datapoint equipment.

Under no circumstances is this guide to be used as a reference in the establishment of specifications or performance criteria. The appropriate product specification, reference manual, or software user's guide should be consulted for that purpose.

Suggestions and additions will be gratefully accepted. Write the Software Support Group, 9725 Datapoint Drive, San Antonio, Texas 78284.

Table of Contents

Device Address Assignments

Device	Binary	Octal
Cassette deck	11110000	360
940X - DSREMOTE #2	11101000	350
940X - DSREMOTE #3	11100100	344
940X - DSREMOTE #4	11100010	342
CRT/Keyboard	11100001	341
940X - DSREMOTE #5	11011000	330
940X - DSREMOTE #6	11010100	324
940X - DSREMOTE #1	11010010	322
9481 - Multifunction Communications Adaptor #1	11010001	321
9481 - Multifunction Communications Adaptor #2	11001100	314
9481 - Multifunction Communications Adaptor #3	11001010	312
9481 - Multifunction Communications	11001001	311
Local Printer #3	11000011	306
Local Printer #2	11000101	305
Local Printer #1	11000110	303
Local Printer #4	10111000	270
Magnetic Tape #1 (Mode 0)	10110100	264
Magnetic Tape #2	10110010	262
Magnetic Tape #3	10110001	261
9483 RIM #4	10101100	254
9483 RIM #5	10101010	252
9483 RIM #6	10101001	251
9404 Synchronous Combox #2	10100110	246
9404 Synchronous Combox #1	10100101	245
9404 Synchronous Combox #3	10100011	243
9483 RIM #1	10011100	234
9483 RIM #2	10011010	232
9483 RIM #3	10011001	231
9420 Parallel Interface #1	10010110	226
9420 Parallel Interface #2	10010101	225
9420 Parallel Interface #3	10010011	223
9426 IBM Channel Simulator	10001110	216
9426 IBM Channel Adaptor	10001101	215
9420 Parallel Interface	10001011	213
Card Reader	10000111	207
9350 Disk Controller	01111000	170
Cartridge Disk	01110100	164
Magnetic Tape (Mode 1 only)	01110010	162
9390 Disk	01110001	161
940X - DSREMOTE #7	01101100	154
940X - DSREMOTE #8	01101010	152
9462 Multiport Adaptor #1	01101001	151
940X - DSREMOTE #9	01100110	146
940X - DSREMOTE #10	01100101	145
940X - DSREMOTE #11	01100011	143
Local Printer #5	01011100	134
Local Printer #6	01011010	132

Device	Binary	Octal
Local Printer #7	01011001	131
Mass Storage Disk	01010110	126
Mass Storage Disk	01010101	125
Mass Storage Disk	01010011	123
Mass Storage Disk #3	01001110	116
Mass Storage Disk #2	01001101	115
9370/9374 Disk Controller	01001011	113
Mass Storage Disk	01000111	107
Diskette #1	00111100	074
Diskette #2	00111010	072
Diskette #3	00111001	071
940X - DSREMOTE #12	00110110	066
940X - DSREMOTE #13	00110101	065
940X - DSREMOTE #14	00110011	063
9462 Multiport Adaptor #4	00101110	056
9462 Multiport Adaptor #2	00101101	055
9462 Multiport Adaptor #3	00101011	053
Unassigned	00100111	047
Unassigned	00011110	036
Unassigned	00011101	035
Unassigned	00011011	033
Unassigned	00010111	027
Unassigned	00001111	017

Note 1: Each communications adaptor will have the device address of the unit it is servicing.

Note 2: Individual disk drives are addressed via the disk controller.

Note 3: Address 0115 is used when converting from 9370 hardware to 9374 hardware or vice versa (using special copyfile overlay).

External Commands

Command - EX (EXPRESSION)

All Devices

COMMAND EXPRESSION	OPERATION NUMBER	COMMAND CODE (OCTAL)	FUNCTION
ADR	1	121	Address Device*
STATUS	2	123	Sense Status*
DATA	3	125	Sense Data*
WRITE	4	127	Write Strobe*
COM1	5	131	Command 1*
COM2	6	133	Command 2*
COM3	7	135	Command 3*
COM4	8	137	Command 4*
$	9	141	$
$	10	143	$
$	11	145	$
$	12	147	$
BEEP	13	151	Beep
CLICK	14	153	Click

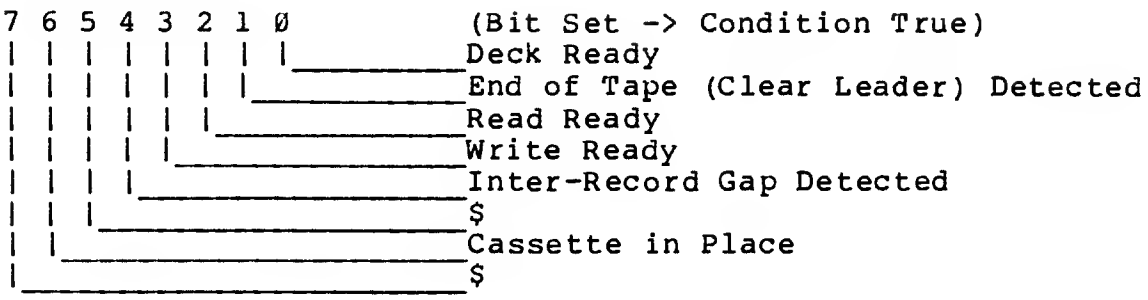
*Note: Also see Debug Command Summary.

Cassette Tape Decks

EXPRESSION	COMMAND NUMBER	OPERATION CODE (OCTAL)	COMMAND FUNCTION
DECK1	15	155	Select Rear Deck
DECK2	16	157	Select Front Deck
RBK	17	161	Read Block
WBK	18	163	Write Block
$	19	165	$
BSP	20	167	Backspace One Block
SF	21	171	Slew Forward
SB	22	173	Slew Backward
REWIND	23	175	Rewind Tape
TSTOP	24	177	Stop Tape

Cassette Tape Decks

Status word - EX STATUS

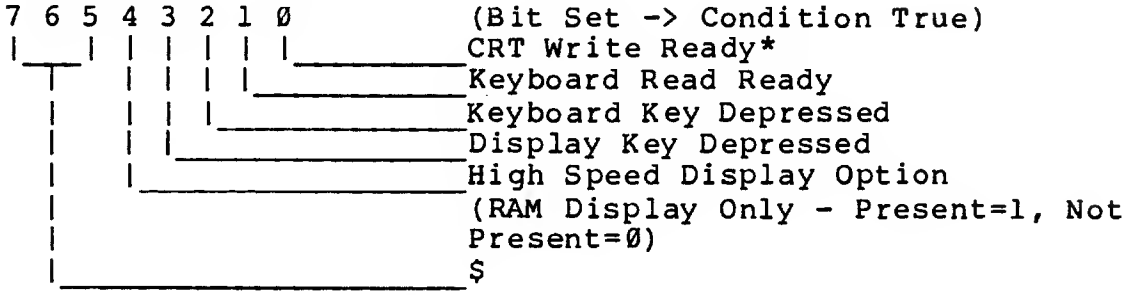


Tape Unit Physical Specifications (ANSI Decks)

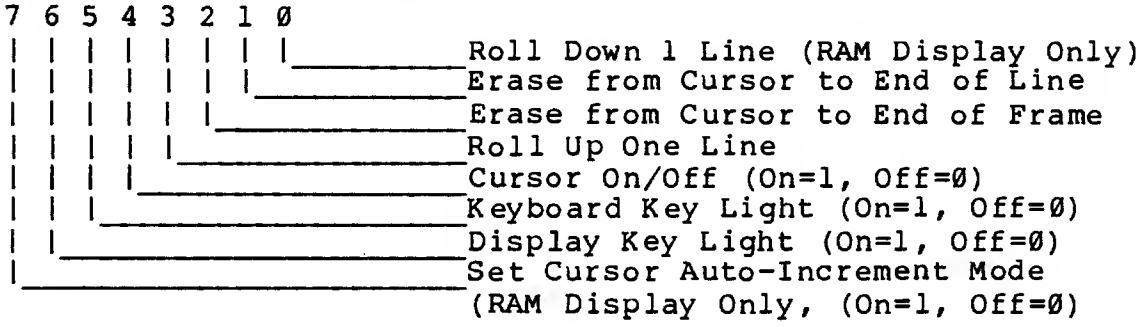
Density	47 characters/inch
Speed	7.5 inches/second
Recording Rate	350 characters/second
Capacity	115,000 characters (typical)
Start/Stop Time (Inter-Record Gap)	305 msec
Start/Stop Distance Inter-Record Gap)	2.2 inches
Rewind Speed	90 inches/second
Rewind Time (maximum)	40 seconds
Characters Transfer Time	2.8 msec

CRT/Keyboard (5500 and 6600 style processors)

Status Word - EX STATUS



Control Word - EX COM1



Control Word - EX COM2

Horizontal Cursor Position
(Decimal 0-79, octal 0-117. Starting at left of screen)

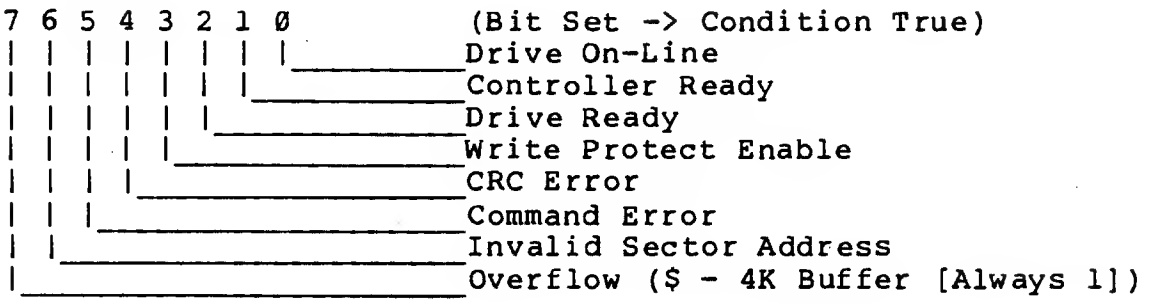
Control Word - EX COM3

Vertical Cursor Position
(Decimal 0-11, octal 0-13. Starting at top of screen)

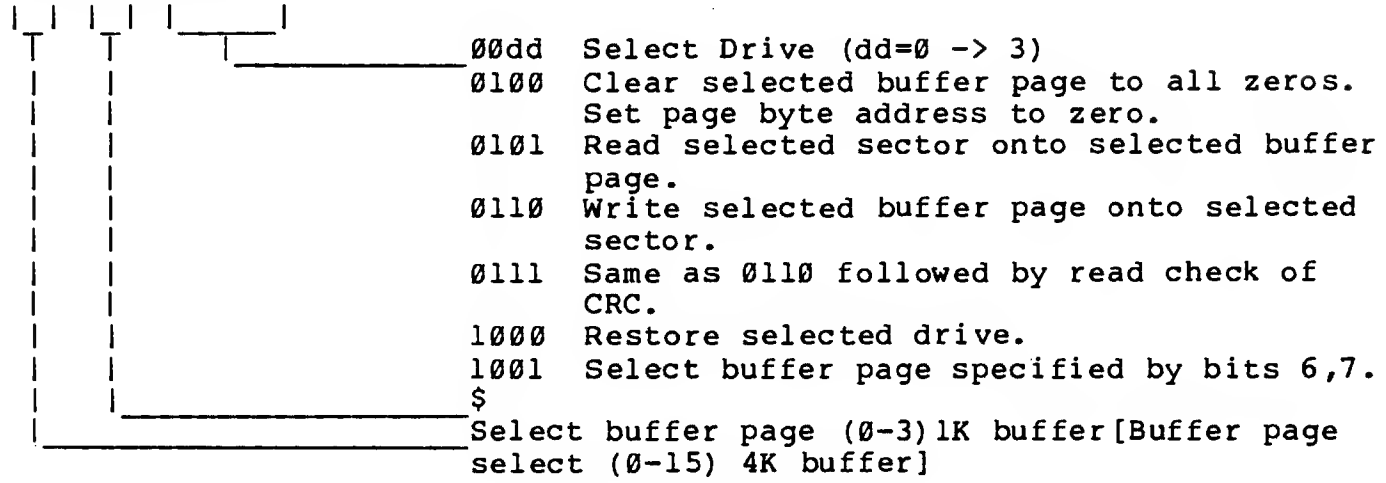
*"Write Ready" is valid only if cursor is positioned to a valid screen position.

9350, 9367 Disk Controller/Drive

Status Word - EX STATUS



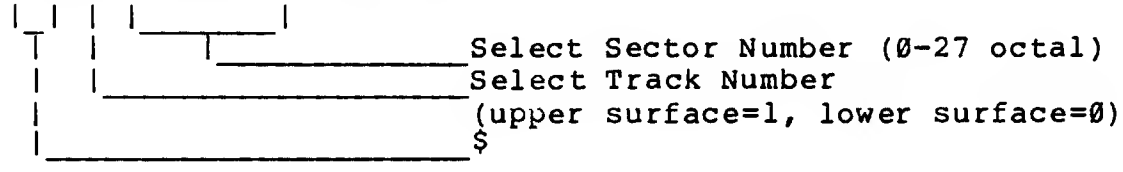
Control Word - EX COM1



Control Word - EX COM2

Select Cylinder Number (0-312 octal)

Control Word - EX COM3



Control Word - EX COM4

Select Buffer Page Byte Address (0-255 decimal, 0-377 octal)

9370, 9374 Disk Controller/Drive

Status Word - EX STATUS

```
7 6 5 4 3 2 1 Ø          (Bit Set -> Condition True)
| | | | | | | |_________Drive On-Line and File Safe (9374 - Drive
| | | | | | |           On-line and No Write check)
| | | | | | |___________Data Transfer in Progress
| | | | | |_____________Drive Busy
| | | | |_______________Seek Incomplete Error
| | | |_________________CRC Error
| | |___________________Write Protect Enable
| |_____________________Sector Not Found
|_______________________Buffer Parity Error
```

Control Word - EX COM1

```
7 6 5 4 3 2 1 Ø
|_____| |_____|
   |       |_____________ØØØØ  Master Clear
   |                      ØØØ1  Disk Read
   |                      ØØ1Ø  Disk Write
   |                      ØØ11  Disk Write Verify (Write followed by read
   |                            check of CRC)
   |                      Ø1ØØ  Restore Selected Drive
   |                      Ø1Ø1  Select Physical Drive as per contents of
   |                            EX COM2 Register (Ø-7)
   |                      Ø11Ø  Select Cylinder as per contents of EX
   |                            COM2 Register (Ø-312 octal)(9374 - Sets
   |                            upper 8 bits of cylinder address
   |                      Ø111  Verify Drive Type: ØØ1 -> Datapoint 9370,
   |                            Ø2Ø -> Datapoint 9374
   |                      1ØØØ  Format Track
   |                      1ØØ1  Select Head as per contents of EX COM2
   |                            Register (Ø-19 decimal, Ø-23 octal) (9374
   |                            - Ø-17 octal)
   |                      1Ø1Ø  Select Sector as per contents of EX COM2
   |                            Register (Ø-24 decimal, Ø-27 octal) 9374
   |                            - Sets upper 5 bits of sector address
   |                      1Ø11  Clear Buffer Parity Error
   |                      11ØØ  Diagnostic Reset: Clear File Unsafe
   |                            (9374 - not used)
   |                      11Ø1  Set Track Offset per contents of EX COM2
   |                            register. (9374 only)
   |_____________________ØØØØ  $
```

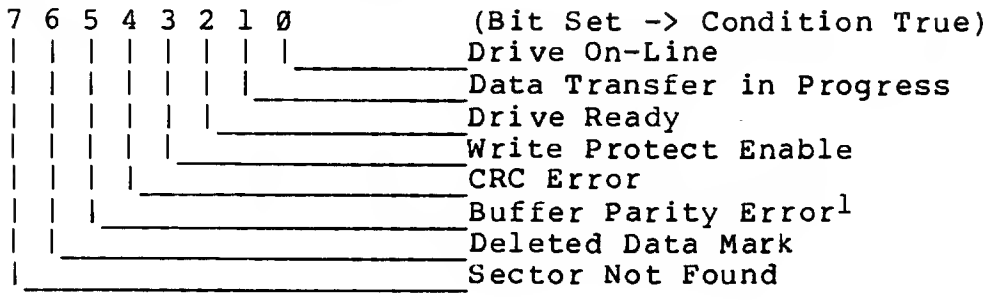
Control Word - EX COM2
  Sets drive cylinder, sector and head in conjunction with proper EX COM1 command. 9374 - also used for track offset selection.

Control Word - EX COM3
  Select buffer page (Ø-15 decimal, Ø-17 octal).

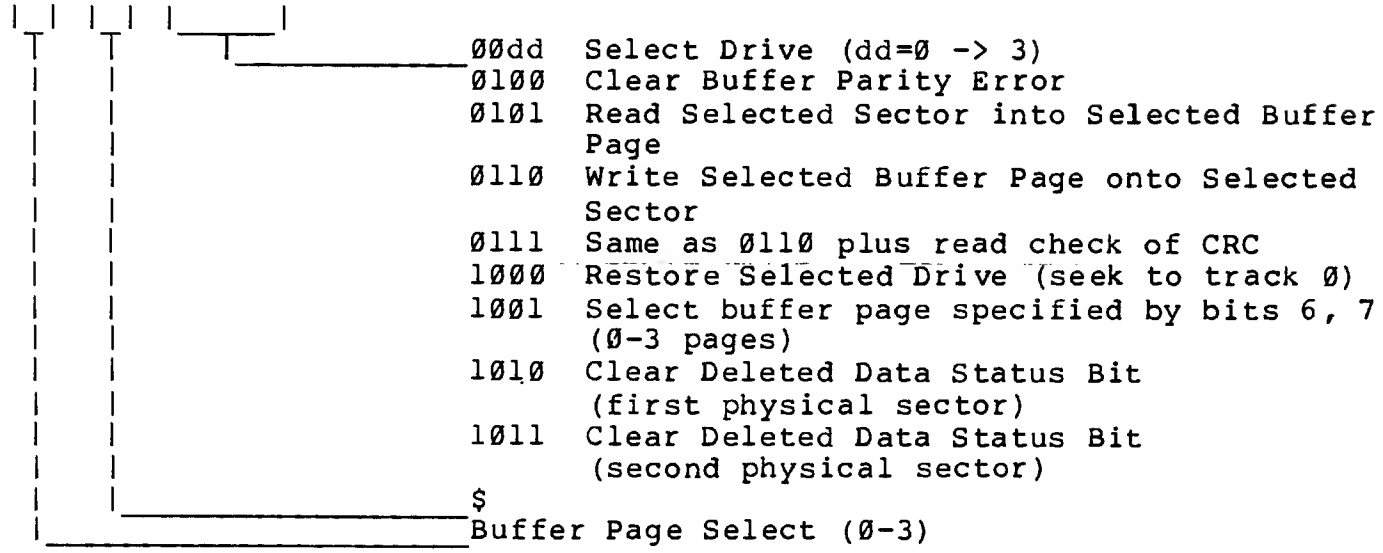
Control Word - EX COM4
  Select buffer page byte address.

9380 Disk Controller/Drive

Status Word - EX STATUS

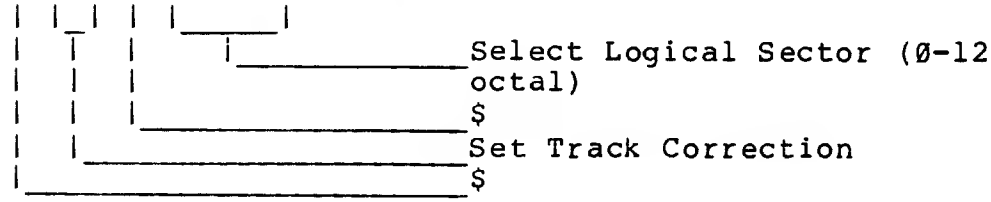


Control Word - EX COM1



Control Word - EX COM2
  Select Track and Seek (Ø-76 decimal, Ø-114 octal)

Control Word - EX COM3

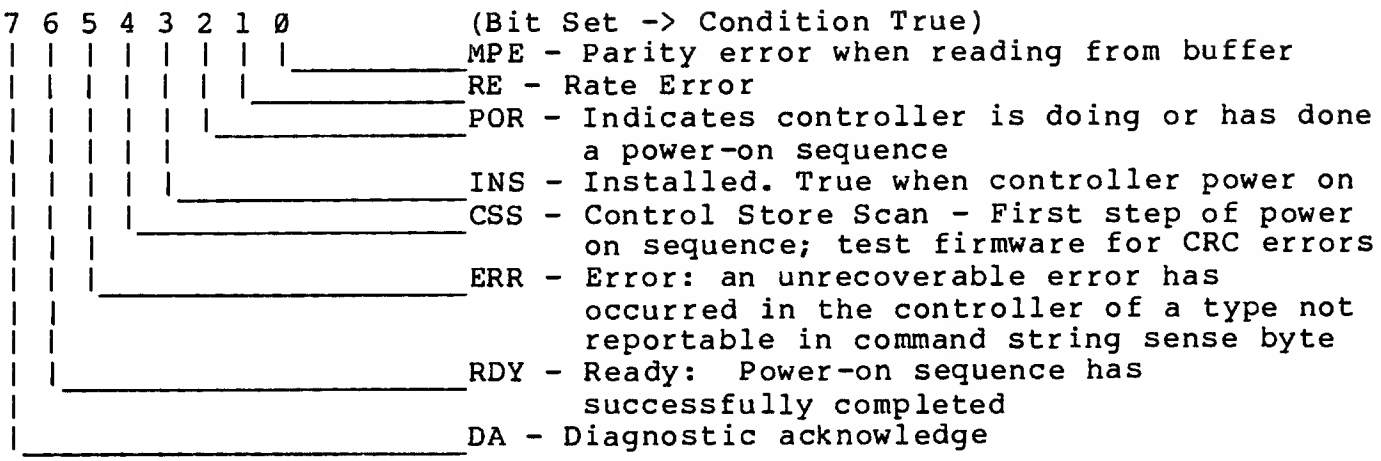


Control Word - EX COM4
  Select Buffer Page Byte Address (Ø-255 decimal, Ø-377 octal)

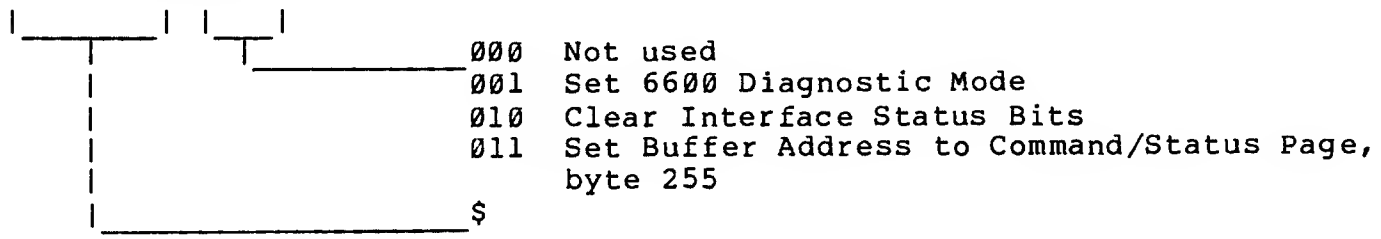
Note 1: Buffer memory parity will be in error on power-up until buffer is written in.
Note 2: The 9380 contains two physical sectors for each logical sector.

9390/9391 Storage Module System

Status Word - EX STATUS



Control Word - EX COM2



Control Word - EX COM3

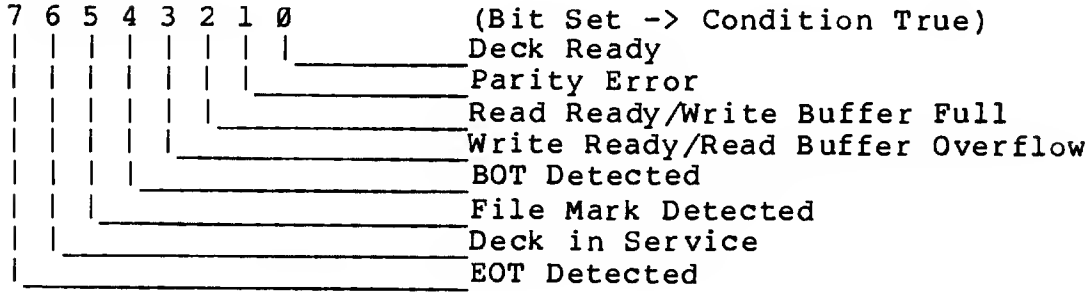
Selects one of 60 buffer memory pages

Control Words - EX COM2, EX COM3, EX COM4

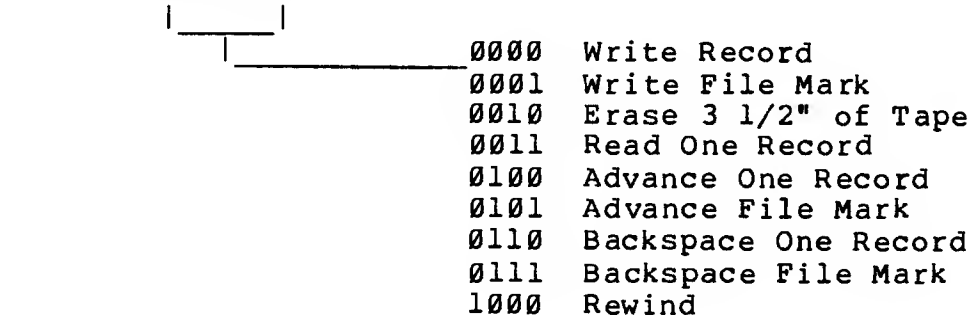
Places the controller in DATA mode

7 and 9 Track Tape Transport (9550/9551, 9552/9553, 9554, 9555)

Status Word - EX STATUS



Control Word - EX COM1



Control Word - EX COM2

Write Buffer Content on Tape

Control Word - EX COM3

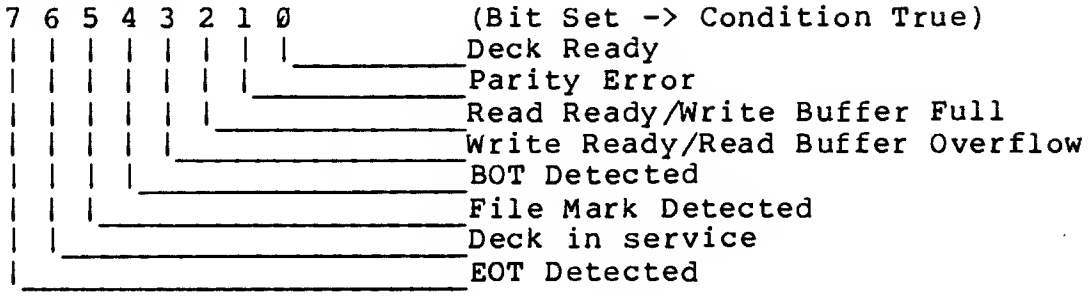
Clear Buffer

Control Word - EX COM4

$

7 Track Tape Transport (9558)

Status Word - EX STATUS



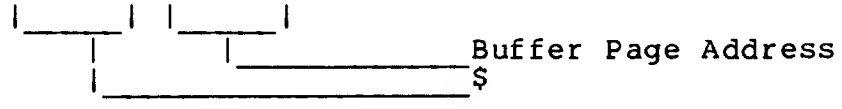
Control Word - EX COM1

7	6	5	4	3	2	1	0	
X	X	X	0	0	0	0	0	Write Record
X	X	X	0	0	0	0	1	Write File Mark
X	X	X	0	0	0	1	0	Erase 3" Tape
X	X	X	0	0	0	1	1	Read Record
X	X	X	0	0	1	0	0	Advance Record
X	X	X	0	0	1	0	1	Advance File Mark
X	X	X	0	0	1	1	0	Backspace Record
X	X	X	0	0	1	1	1	Backspace File Mark
X	X	X	0	1	0	0	0	Rewind
0	0	0	0	1	0	0	1	Drive 0 Select
0	1	0	0	1	0	0	1	Drive 1 Select
1	0	0	0	1	0	0	1	Drive 2 Select
1	1	0	0	1	0	0	1	Drive 3 Select
X	X	X	0	1	0	1	0	Slew Write
X	X	X	0	1	0	1	1	Slew Read
X	X	X	0	1	1	0	0	Slew Halt
X	X	X	0	1	1	0	1	Load Write Pointer
X	X	X	0	1	1	1	0	Load Read Pointer
X	X	X	1	0	0	0	0	Write Edit
X	X	X	1	0	1	1	0	Backspace Edit

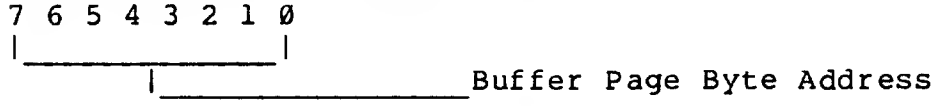
Control Word - EX COM2
  Write Buffer Content on Tape

Control Word - EX COM3
  Clear Buffer

Control Word - EX COM4 (First Output)

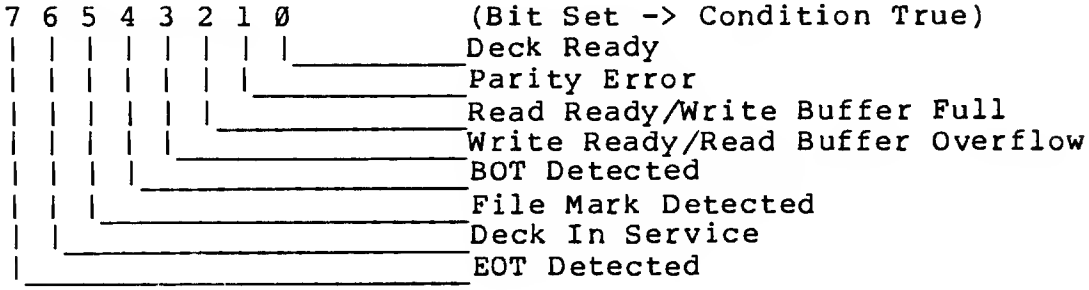


Control Word - EX COM4 (Second Output)

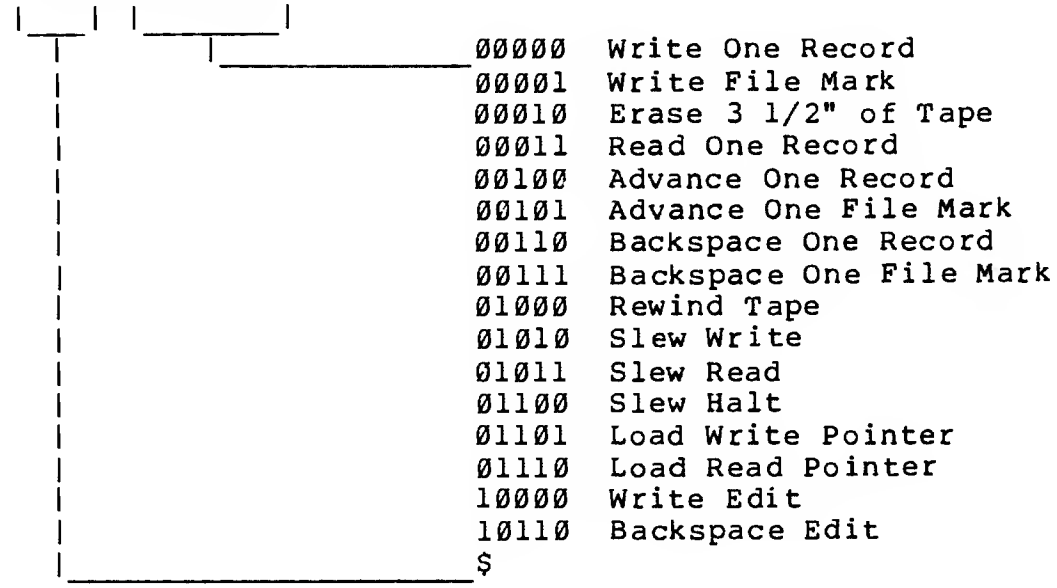


1600 BPI Magnetic Tape System (9580, 9581, 9583)

Status Word - EX STATUS



Control Word - EX COM1

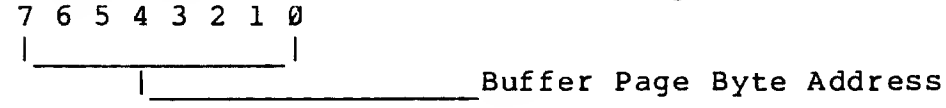


Control Word - EX COM2
Write Buffer Contents on Tape

Control Word - EX COM3
Clear Buffer

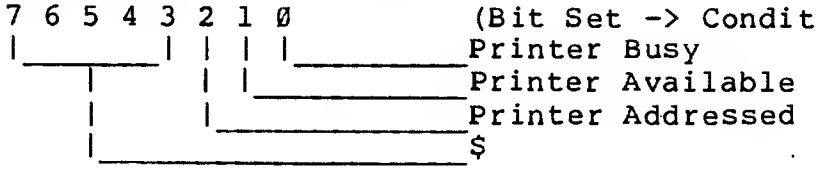
Control Word - EX COM4 (First Output)

Control Word - EX COM4 (Second Output)



Printers (9212/9214, 9242, 9260, 9265, 9280, 9291, 9292)

Status Word - EX STATUS



Character Transmission - EX WRITE
See Character Transmission and Translation Table

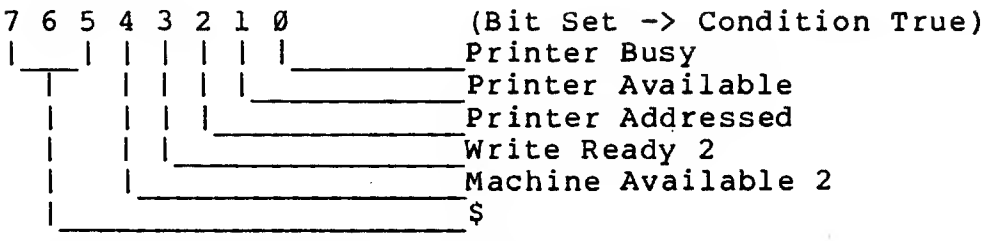
Printer Control Codes (in octal)

012	Line Feed
013	Vertical Tab (Centronics only)
014	Form Feed
015	Print
016	Elongated Print (Centronics only)

Note 1: Printer Control Codes are in addition to printing the buffer (i.e. data followed by a 014 will print the buffer and form feed).

Freedom Printer (9232/9234)

Status Word - EX STATUS



Character Transmission - EX WRITE
See "Character Transmission and Translation Table"

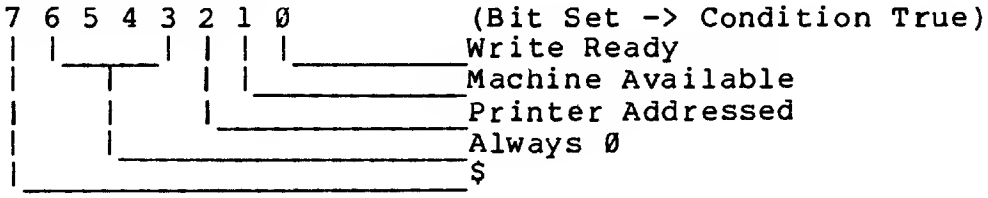
Printer Control Codes[1] (in octal)

Ø01	Tab to column (MSB, LSB)
Ø02	Secondary tractor line feed
Ø05	Set secondary tractor left margin and tab to column (MSB, LSB)
Ø12	Line feed
Ø13	Vertical tab
Ø14	Form feed
Ø15	Print
Ø16	Primary tractor micro-line feed
Ø36	Secondary tractor micro-line feed

Note 1: Printer Control Codes are in addition to printing the buffer (i.e., data followed by a Ø14 will print the buffer and form feed).

Printers (9601/9602, 9621/9622, 9257/9258)

Status Word - EX STATUS



9601/9602 Control Codes

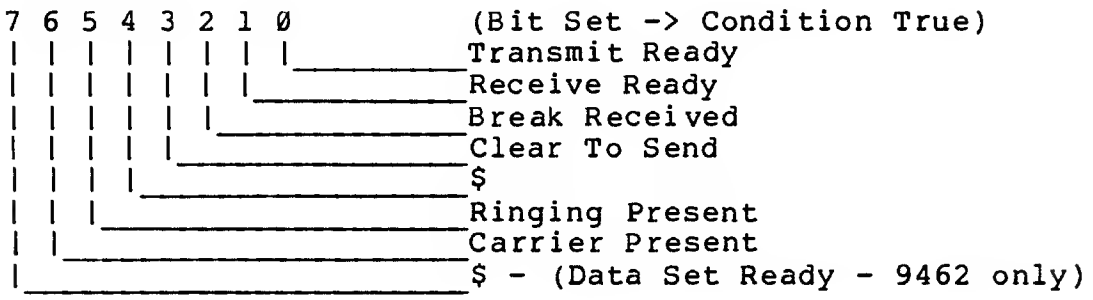
Initialize	033 0143
Default	033 0121
Set Form Length	033 0133 n 073 x 0162
Set Left Margin	033 0133 n 0161
Set Spacing	033 0133 h 073 n 040 0107
Load Character Table	033 0120 040 n 073 p 073 040 e 073 040 w 073 041 e 073 041 w 073 ... 0176 e 073 0176 w 073 0177 e 073 0177 w 033 0134
Pause	033 0120 044 MSG 033 0134
Vertical Tab	033 133 n 0145
Micro Line Feed	016
Line Feed	012
Form Feed	014
Thin Space	033 0133 n 0141
Space	040
Horizontal Tab	033 0133 n 0140
Carriage Return	015
Print	040 033 0116 041
Print	0177 033 0116 042
Bell	007
Delete	0177
Printer On	032
Printer Off	024

9621/9622 Control Codes

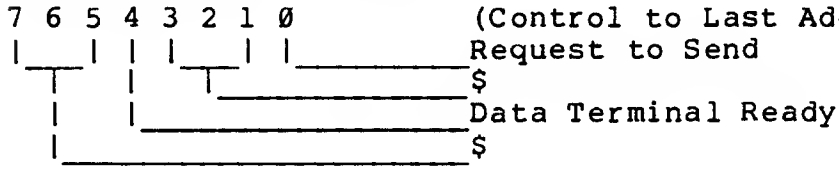
Carriage Return	015
Tab to Column	001 msb lsb
Line Feed	012
Micro Line Feed	016
Form Feed	014
Downline Load Forms Length	033 014 n2 n1
Forms Length Reset	023
Select a Resident Font	033 0133 #a 0155
Horizontal Tab Set	033 0120 042 #1 ... #i 033 0134
Vertical Tab Set	033 0120 043 #1 ... #i 033 0134

Multiple Port Communications Adaptor (9460, 9462)

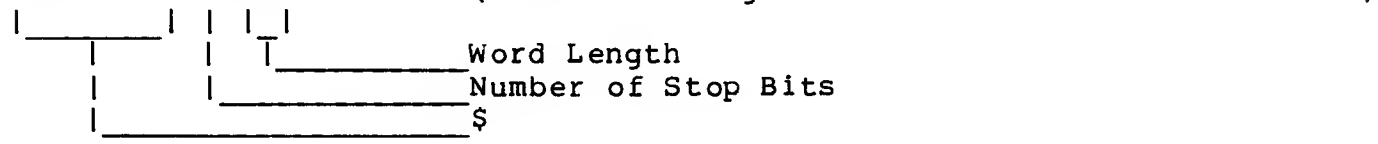
Status Word[1] - EX STATUS



Control Word - EX COM1



Control Word - EX COM2



Transmit and Receive Character Formats

Control Bit Position 210-Octal	Start Bit	Information Bits	Stop Bits	Code Bit Positions 7 6 5 4 3 2 1 0
000-0	1	5	1	x x x 5 4 3 2 1
001-1	1	6	1	x x 6 5 4 3 2 1
010-2	1	7	1	x 7 6 5 4 3 2 1
011-3	1	8	1	8 7 6 5 4 3 2 1
100-4	1	5	2	x x x 5 4 3 2 1
101-5	1	6	2	x x 6 5 4 3 2 1
110-6	1	7	2	x 7 6 5 4 3 2 1
111-7	1	8	2	8 7 6 5 4 3 2 1

Control Word - EX COM3

Control Word - EX COM4 - Not used

Control Word - EX WRITE

Transfers character in A register to the currently selected port for transmission.

Note 1: Until a port is selected, the STATUS word is a logical OR value of the status of all eight ports.

Multi Function Communications Adaptor (9481)

Status Word - EX STATUS

7	6	5	4	3	2	1	0	(Bit Set -> Condition True)
0	0	0	0	0	0	0	0	Set normal status mode
0	0	1	0	0	0	0	0	Set transmit buffer status mode
1	0	0	0	0	0	0	0	Set modem status mode
1	1	0	0	0	0	0	0	Set Receive Status Mode
0	0	1	1	0	0	0	0	Set ACU Status Mode
0	0	1	1	0	0	0	1	Clear Transmit Buffer
0	0	1	1	0	0	1	0	Set BSC Mode
0	0	1	1	0	0	1	1	Reset Line Active Status
0	0	1	1	1	n	n	n	Set Generalized Synchronous Mode
					n	n	n	Character length (bits)
					0	0	0	8
					0	0	1	8
					0	1	0	8
					0	1	1	8
					1	0	0	4
					1	0	1	5
					1	1	0	6
					1	1	1	7

SDLC Normal Status Byte

7	6	5	4	3	2	1	0	
_	_	_	_	_	_	_	_	
|	|	|	|	|	|	|	|	
|	|	|	|	|	|	|	|	Receive Ready
|	|	|	|	|	|	|		Line Active
|	|	|	|	|	|			$
|	|	|	|	|				ØØ Character is data character.

01 Character is an SDLC FLAG (0176) but the preceding two CRC bytes indicate an error occurred in the previous frame.

1Ø Character is a control character other than FLAG.

11 Character is an SDLC FLAG and the two preceding CRC bytes were valid, indicating that no error occurred.

BSC Normal Status

7	6	5	4	3	2	1	0	
|	|	|	|	|	|	|	|	
|	|	|	|	|	|	|	|	Receive Ready
|	|	|	|	|	|	|		$
|	|	|	|	|	|			End of Block (EOB)
|	|	|	|	|				CRC Good

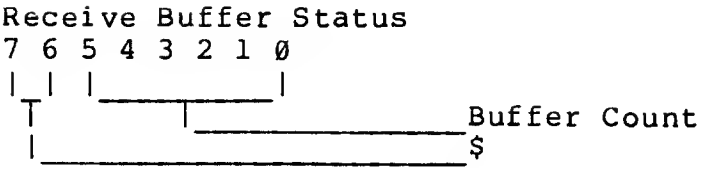
Generalized Synchronous Normal Status Byte

7	6	5	4	3	2	1	0	
|	|	|	|	|	|	|	|	
|	|	|	|	|	|	|	|	Receive Ready
|	|	|	|	|	|	|		$

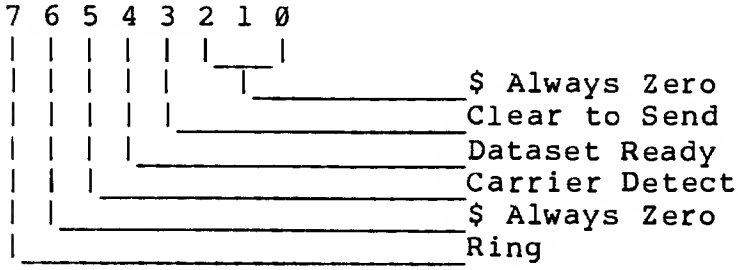
Transmit Buffer Status

7	6	5	4	3	2	1	0	
_	_	_	_	_	_	_	_	
|	|	|	|	|	|	|	|	
|	|	|	|	|	|	|		Buffer Positions Available
|	|	|	|	|	|			$

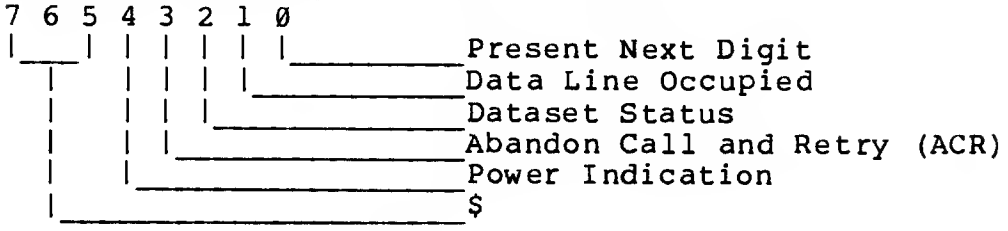
Receive Buffer Status



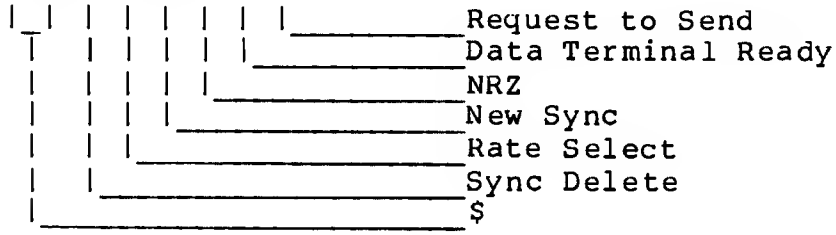
Modem Status



Automatic Calling Unit (ACU) Status



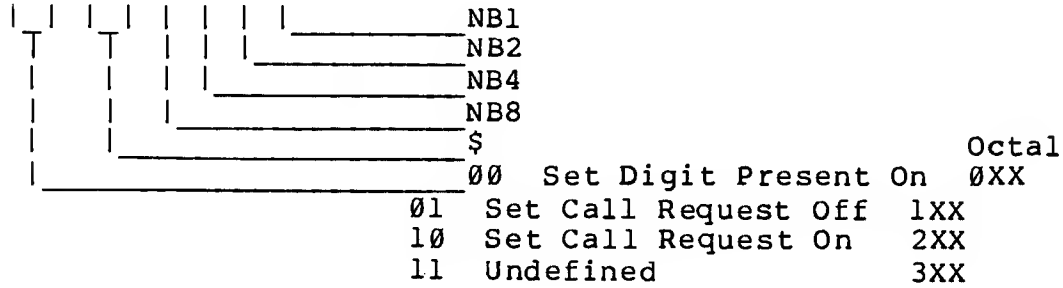
Control Word - EX COM1



Control Words - EX COM2 and EX WRITE

Transfer data to buffer. EX COM2 is used to insert special control codes into the data stream.

Control Word - EX COM3



Dial Digit NB1 - NB8 (bits Ø - 3)

Dial Digit	NB8	NB4	NB2	NB1
Ø	1	1	1	1
1	1	1	1	Ø
2	1	1	Ø	1
3	1	1	Ø	Ø
4	1	Ø	1	1
5	1	Ø	1	Ø
6	1	Ø	Ø	1
7	1	Ø	Ø	Ø
8	Ø	1	1	1
9	Ø	1	1	Ø
EON	Ø	Ø	1	1
SEP	Ø	Ø	1	Ø

EON (ØØ3) End of Number Code
SEP (ØØ2) Separator or inter-digit relay code

Control Word - EX COM4
Used to set the receive and transmit rate.

Asynchronous Communications Adaptors

Status Word - EX STATUS (Bit set -> Condition True)

Status Word	Bits	9400	9401	9402	9403	9410
Ø		Transmit Ready	Same	Same	Same	Present Next Digit
1		Receive Ready	Same	Same	Same	Data Line Occupied
2		Break Received	Same	Same	Same	Distant Station Connected
3		Clear to Send (CB)	Same	Same	$	Abandon Call
4		Reverse Carrier Present	$ (SB)	Same	$	Power Indication
5		Ringing Present	Same (CE)	Same	$	Standby Indicator
6		Main Carrier Present (CF)	Carrier Present (CAR)	Same	$	$
7		$	Data Coupler Ready (DCR)	Data Coupler Ready (DCR)	$	$

Control Word - EX COM1

EX COM1 Word Bits	9400	9401	9402	9403	9410
0	Request to Send	Same	Same	$	Data Terminal Ready
1	Invert Received Data Line	Same	Same	$	Request to Send Mn. Channel
2	Supervisory Channel On	$	Same	$	Sig. Rate Sel./Transmit Freq. Sel.
3	Invert Transmitted Data Line	Same	Same	Same	Select Standby
4	Data Terminal Ready	Off Hook	Off Hook	$	Receiver Cut-off Mn. Channel
5	$	$	Send 2025 Hz	$	Return to Non-Data Mode
6	$	Orig.=1[1] Ans.=0	Orig.=1 Ans.=0	$	Request to Send Rev. Channel
7	$	Send Dial Pulses	Send Dial Pulses	$	Receiver Cut-off Rev. Channel

Control Word - EX COM2 (1st execution)
Receive Time Base - least significant byte (see charts)

Control Word - EX COM2 (2nd execution)
Receive Time Base - most significant byte (see charts)

Control Word - EX COM3 (1st execution)
Transmit Time Base - least significant byte (see charts)

Control Word - EX COM3 (2nd execution)
Transmit Time Base - most significant byte (see charts)

Note 1: 2025 Hz is used for transmission if this bit is 0.

Time Base Chart (EIA) 9400, 9401, 9402, 9403

Receive	EX COM2	EX COM2
Transmit	EX COM3	EX COM3

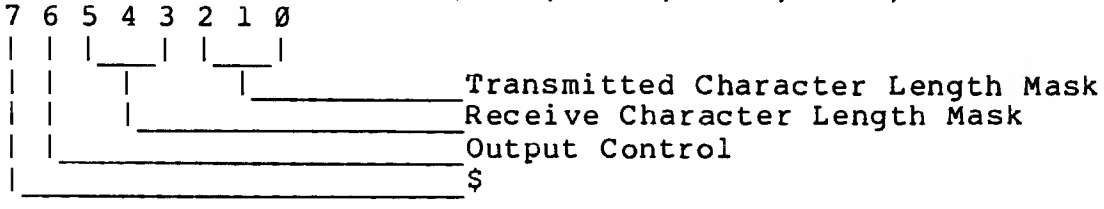
Bit Rate	1st Mask Word (Octal)	2nd Mask Word (Octal)
100	375 (dialing)	000
110	375	106
220	376	243
440	377	121
150	376	000
300	377	000
600	377	200
1200	377	300
1800	377	325
2400	377	340

Time Base Chart (CCITT) 9410

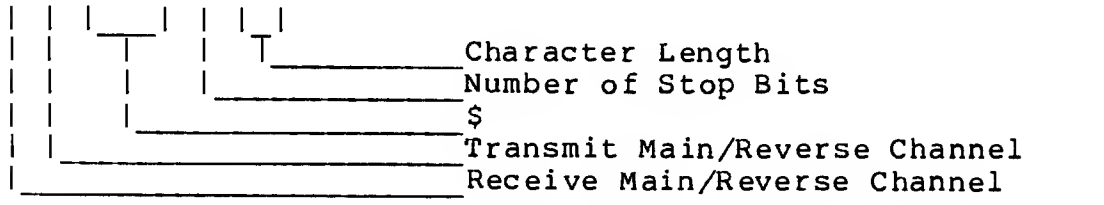
Receive	EX COM2	EX COM2
Transmit	EX COM3	EX COM3

Bit Rate	1st Mask Word (Octal)	2nd Mask Word (Octal)
50	030	000
75	020	000
100	014	000
200	006	000
110	365	027
220	372	214
440	375	106
150	370	000
300	374	000
600	376	000
1200	377	000
2400	377	200
4800	377	300
9600	377	340

Control Word - EX COM4 (9400, 9401, 9402, 9403)

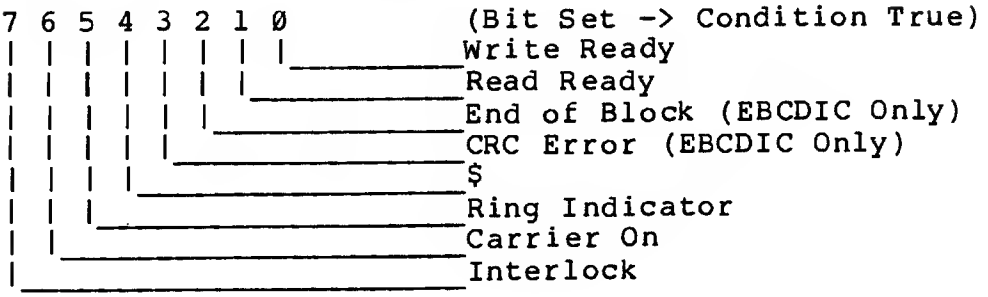


Control Word - EX COM4 (9410)

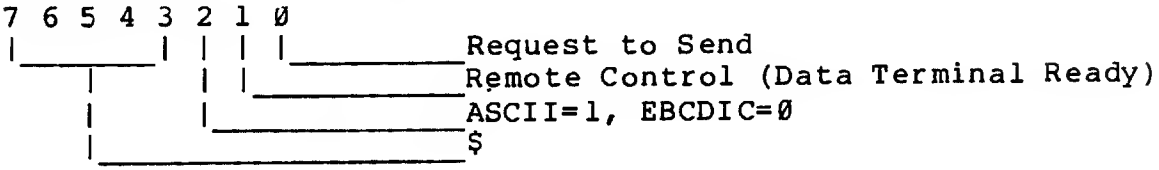


9Ø4 Synchronous Communications Adaptor

Status Word - EX STATUS



Control Word - EX COM1



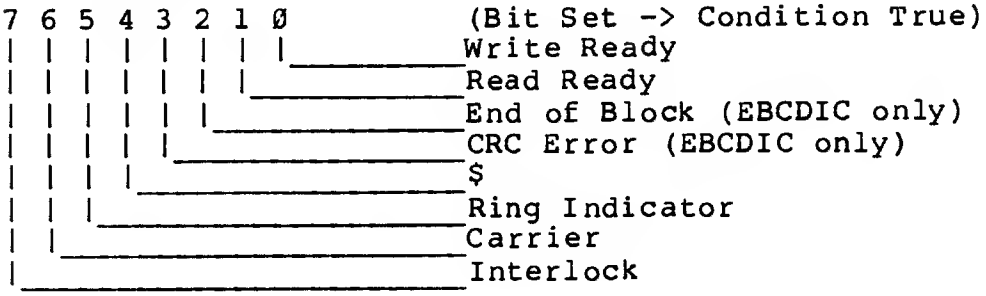
Control Word - EX COM2
  Write single "DLE" ("DLE" character loaded in the A register, EBCDIC only)

Control Word - EX COM3
  New SYNC

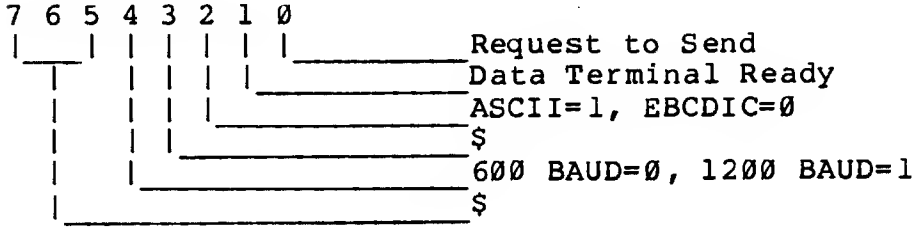
Control Word - EX COM4
  $

9405 Synchronous Communications Adaptor (9404 with 712 clock option)

Status Word - EX STATUS



Control Word - EX COM1



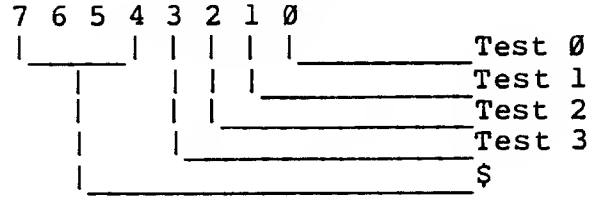
Control Word - EX COM2

Write single "DLE" ("DLE" character loaded in the A register, EBCDIC only)

Control Word - EX COM3

New SYNC

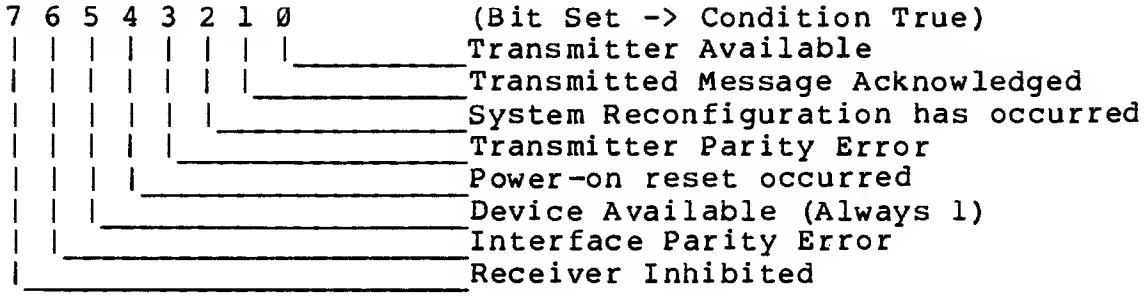
Control Word - EX COM4



Note 1: The toggle switch on the 9405 board must be flipped for 600/1200 BAUD operation.

9483 Resource Interface Module

Status Word - EX STATUS



Control Word - EX COM1

General RIM command defined by the contents of the processor output bus.

Control Word - EX COM4

Sets buffer address to the contents of the processor output bus and sets DATA mode in the RIM. All three page registers remain unchanged. The next EX WRITE or INPUT instruction executed will access the location in the buffer memory specified by this address and the processor page register.

1800/3800 Serial Interface Module

```
SISTART (165)           Serial Interface START
SISYNC (111 165)        Serial Interface SYNChronize
SIIN (163)              Serial Interface IN
SIMIN (062 163)         Serial Interface Multiple In
SICOUT (167)            Serial Interface Control OUT
SICMOUT (111 167)       Serial Interface Control Multiple OUT
SIOUT (062 167)         Serial Interface OUT
SIMOUT (113 167)        Serial Interface Multiple OUT
```

```
SIMODIN (161)           Serial Interface MODem IN
7 6 5 4 3 2 1 0
| | | | | |___|
| | | | |   |_____________$
| | | | |_________________Clear to Send
| | | |___________________Data Set Ready
| | |_____________________Received Line Signal Detector
| |_______________________Ring Indicator
|_________________________$
```

```
SIMODOUT (062 161)      Serial Interface MODem OUT
7 6 5 4 3 2 1 0
|___| | | | | |_________Request to Send
  |   | | | |___________Data Terminal Ready
  |   | | |_____________$
  |   | |_______________New Sync/Rate Select
  |   |_________________Break/Spare
  |_____________________$
```

```
SIACUIN (111 161)       Serial Interface ACU IN
7 6 5 4 3 2 1 0
|_| | | | | | |_________Present Next Digit/Secondary Signal Detect
 |  | | | | |___________Data Line Occupied/Secondary Clear to Send
 |  | | | |_____________Call Origination Status/Secondary Received Data
 |  | | |_______________Abandon Call and Retry/Signal Quality Detector
 |  | |_________________Power indication
 |  |___________________Spare
 |_____________________$
```

```
SIACUOUT (113 161)      Serial Interface ACU OUT
7 6 5 4 3 2 1 0
|_| | | | | | |_________Digit 1
 |  | | | | |___________Digit 2
 |  | | | |_____________Digit 4
 |  | | |_______________Digit 8
 |  | |_________________Digit Present/Secondary Transmitted Data
 |  |___________________Call Request/Secondary Request to Send
 |_____________________$
```

2200 Machine Instructions (SNAP)

Function	Operation Code	Description
Lr_d (exp)	$0n_d6$,vvv	Load immediate
Lr_dr_s	$3n_dn_s$	Load
AD (exp)	004,vvv	Add immediate
ADr_s	$20n_s$	Add
AC (exp)	014,vvv	Add with carry immediate
ACr_s	$21n_s$	Add with carry
SU (exp)	024,vvv	Subtract immediate
SUr_s	$22n_s$	Subtract
SB (exp)	034,vvv	Subtract with borrow immediate
SBr_s	$23n_s$	Subtract with borrow
ND (exp)	044,vvv	And immediate
NDr_s	$24n_s$	And
OR (exp)	064,vvv	Or immediate
ORr_s	$26n_s$	Or
XR (exp)	054,vvv	Exclusive - or immediate
XRr_s	$25n_s$	Exclusive - or
CP (exp)	074,vvv	Compare immediate
CPr_s	$27n_s$	Compare
SRC	012	Shift right circular
SLC	002	Shift left circular
JMP (adr)	104,lsb,msb	Unconditional jump
JTc (adr)	1p0,lsb,msb	Jump on true flag
JFc (adr)	1m0,lsb,msb	Jump on false flag
CALL (adr)	106,lsb,msb	Unconditional call
CTc (adr)	1p2,lsb,msb	Call on true flag
CFc (adr)	1m2,lsb,msb	Call on false flag
RET	007	Unconditional return
RTc	0p3	Return on true flag
RFc	0m3	Return on false flag
BETA*	020	Register and F/F mode swap
ALPHA*	030	Register and F/F mode swap
PUSH*	070	Address onto stack
POP*	060	Address from stack
DI*	040	Disable interrupt
EI*	050	Enable interrupt
NOP	300	No operation
HALT	377	Halt
INPUT	101	Input
EX	See tables	External command

r_s=source register
r_d=destination register
(exp)=one-byte expression
(adr)=two-byte address
c=condition flag
n_d,n_s=register reference number
vvv=expression value
lsb=least significant byte of address
msb=most significant byte of address

*Version II 2200 only
1. Add 1.6 usec if memory reference
2. Add 1.6 usec if transfer occurs
p,m=condition code reference

Register Reference Table

r	n
A	Ø
B	1
C	2
D	3
E	4
H	5
L	6
X*	7
M	7

*55ØØ only

M=memory reference. Memory location specified by HL* (or currently selected register pair).

Condition (Flip-flop) Code Reference Table

c	m	p
C (Carry)	Ø	4
Z (Zero)	1	5
S (Sign)	2	6
P (Parity) (true= odd parity)	3	7

Register Codes (55ØØ only)

r	[r] (register select opcode)	[pr] (register select for paged instructions)
A	no code (implicit)	1Ø5
B	111	114
C	Ø62	124
D	113	134
E	174	144
H	115	154
L	176	164
X	117	no code (cannot be used)

Register Pair Codes (55ØØ only)

rp	[rp] - register pair select code
HL	176 (implicit)
BC	Ø62
DE	174
XA	Ø22

5500 Machine Instructions (SNAP/3)

Function	Operation Code	Description
L(rd)M		
L(rd)M (rp)	[rp],$3n_d7$	Load register from memory, memory address in rp
LM(rs)		
LM(rs) (rp)	[rp],$37n_s$	Load memory from register memory address in rp
L(rd) (rs)	$3n_dn_s$	Load
L(r)data	Ønd6,vvv	Load immediate
AD(rs)	20ns	Add
AC(rs)	21ns	Add with carry
SU(rs)	22ns	Subtract
SB(rs)	23ns	Subtract with borrow
ND(rs)	24ns	And
XR(rs)	25ns	Exclusive Or
OR(rs)	26ns	Or
CP(rs)	27ns	Compare
AD(rs)(rd)		
AC(rs)(rd)		
SU(rs)(rd)		
SB(rs)(rd)		
ND(rs)(rd)		
XR(rs)(rd)		
OR(rs)(rd)		
CP(rs)(rd)		
ADM		
ACM		
SUM		
SBM		
NDM		
XRM		
ORM		
CPM		
ADM(rd)		
ACM(rd)		
SUM(rd)		
SBM(rd)		
NDM(rd)		
XRM(rd)		
ORM(rd)		
CPM(rd)		
AD data	Ø04,vvv	Add immediate
AC data	Ø14,vvv	Add with carry immediate
SU data	Ø24,vvv	Subtract immediate
SB data	Ø34,vvv	Subtract with borrow immediate
ND data	Ø44,vvv	And immediate

Function	Operation Code	Description
XR data	Ø54,vvv	Exclusive or immediate
OR data	Ø64,vvv	Or immediate
CP data	Ø74,vvv	Compare immediate
AD(r) data		
AC(r) data		
SU(r) data		
SB(r) data		
ND(r) data		
XR(r) data		
OR(r) data		
CP(r) data		
SLC	Ø02	Shift left circular
SRC	Ø12	Shift right circular
SRE	Ø32	Shift right extended
SLC(r)	[rd],Ø02	Shift left circular, other than A reg.
SRC(r)	[rd],Ø12	Shift right circular, other than A reg.
SRE(r)	[rd],Ø32	Shift right extended, other than A reg.
JMP loc	1Ø4,lsb,msb	Unconditional jump
Jcc loc		
Jcc loc (fall thru)		
EJMP loc		
NOJ loc	Ø45	NOP jump, skip next two bytes
NOP	3Ø0	No operation
CALL loc	1Ø6,lsb,msb	Unconditional call
Ccc loc		
Ccc loc (fall thru)		
RET	Ø07	Unconditional return
Rcc		
Rcc (fall thru)		
UR		
EUR		
IN		
IN(r)		
PIN	1Ø3	Input w/parity testing
PIN(rd)	[rd],1Ø3	Input w/parity testing, to other than A register
EX ADR		
EX(r)ADR		
EX (exp)		
EX(r) (exp)		
EX STATUS		
EX(r)STATUS		
EX DATA		
EX(r) DATA		
EX WRITE		

Function	Operation Code	Description
EX(r)WRITE		
EX COM1		
EX(r)COM1		
EX COM2 & 3		
EX(r) COM2 & 3		
EX COM4		
EX(r) COM4		
MIN	111,061	Multiple In, DMA-type command. I/O device to memory starting at HL.
MOUT	111,071	Multiple Out, DMA-type command. I/O device from memory starting at HL.
BETA	020	Register and F/F Mode Swap
BETA (in Beta)		
ALPHA	030	Register and F/F Mode Swap
ALPHA (in Alpha)		
DI	040	Disable Interrupt
EI	050	Enable Interrupt
POP	060	Address from Stack
POP (rp)	[rp],060	Address from stack, into rp
PUSH	070	Address onto Stack
PUSH (rp)	[rp],070	Address onto stack, from rp
PUSH loc		
BT (B=0)	021	Block transfer
BT (B≠0)		
BTR (B=0)	111,021	Block transfer reverse
BTR (B≠0)		
BCV (B=0)		
BCV (B≠0)		
BCP (B=0)	041	Block Compare
BCP (B≠0)		
BFAC	011	Binary field add with carry
BFSB	031	Binary field subtract with borrow
DFAC	111,041	Decimal field add with carry
DFSB	062,041	Decimal field subtract with borrow
BFSL	075	Binary field shift left
BFSR	111,075	Binary field shift right
STKS	065	Stack store, save stack in memory
STKL	111,065	Stack load, restore stack from memory
REGS	055	Register store, save registers in memory, descending from top of stack address
REGL	111,055	Register load, restore registers from memory, descending from address in HL
CCS		
CCS (rd)	[rd],042	Condition code save in rd, add rd to itself to restore conditions
		Increment Register Pair Instructions

Function	Operation Code	Description
INCP HL	Ø15	HL by 1
INCP HL,A	Ø17	HL by contents of A
INCP (rp)		
INCP (rp),2		
INCP (rp),A		
INCP XA	Ø22,Ø15	XA by 1
INCP XA,2	111,Ø15	XA by 2
INCP XA,A	Ø22,Ø17	XA by contents of A
		Decrement Register Pair Instructions
DECP HL	Ø35	HL by 1
DECP HL,A	Ø37	HL by contents of A
DECP (rp)		
DECP (rp),2		
DECP (rp),A		
DECP XA	Ø22,Ø35	XA by 1
DECP XA,2	111,Ø35	XA by 2
DECP XA,A	Ø22,Ø37	XA by A
DL DE,HL	Ø47	
DL BC,HL	111,Ø47	
DL BC,BC	Ø62,Ø47	
DL BC,DE	113,Ø47	
DL DE,BC	174,Ø47	
DL DE,DE	115,Ø47	
DL HL,BC	176,Ø47	
DL HL,DE	117,Ø47	
DL HL,HL	Ø57	
DS DE,HL	Ø27	
DS BC,HL	111,Ø27	
DS BC,DE	113,Ø27	
DS DE,BC	174,Ø27	
DS HL,BC	176,Ø27	
DS HL,DE	117,Ø27	
PL (r),loc		
PS (r),loc		
DPL (rp),loc		
DPS (rp),loc		
INCI(dsp),Ø05,lsb,[i] (idx)		Increment index by lsb of (dsp)
DECI(dsp), (idx)	Ø25,lsb,[i]	Decrement index by lsb of (dsp)
INCI*(dsp), (idx)	111,Ø05,lsb, msb,[i]	Increment index by msb,lsb of (dsp)
DECI*(dsp), (idx)	111,Ø25,lsb, msb,[i]	Decrement index by msb,lsb of (dsp)
LFII(rp),	[rp],Ø05,lsb,	Load from index incremented, add lsb of (dsp) to

Function	Operation Code	Description
(dsp),(idx)	[i]	index value and save result in rp (does not modify value in index)
LFID(rp), (dsp),(idx)	[rp],005,lsb, [i]	Load from index decremented, subtract lsb from index and save result in rp (does not modify value in index).
LFII(rp),* (dsp),(idx)	[srp],005,lsb, msb,[i]	Load from index incremented, same as LFII above, but using msb,lsb of (dsp)
LFID(rp),* (dsp),(idx)	[srp],025,lsb, msb,[i]	Load from index decremented, same as LFID above, but using msb,lsb of (dsp).
BRL		
BRL(r)		
STL	077	Sector table load
SC	067	System Call, call 0167452
BP	052	Break Point, call 0167460 (DEBUG)
HALT	377	Halt
HALT (user mode)		
QQQQ (undefined)		
QQQQ(r)		
EX BEEP		See external command tables
EX BEEP (in progress)		
EX CLICK		
EX CLICK (in progress)		
Sync (010)		
DPLR		
DPSR		
STLOr		
INFO		
INFO2	062,010	
INFO3	113,010	
INFO4	174,010	
INFO5	115,010	
INFO6	176,010	
INFO7	117,010	
INFO8	022,010	
		System Status Information
SYSTAT1	111,157	
SYSTAT2	062,157	
SYSTAT3	113,157	
SYSTAT4	174,157	
SYSTAT5	115,157	
SYSTAT6	176,157	
SYSTAT7	117,157	
SYSTAT8	022,157	
LODCF	155	Load character font
ACDO	rp 151	Perform audio

Function	Operation Code	Description
ACDOO	rp 153	Perform audio override
ACCGP		
SYSSAV	062,020	
ALPHAL	111,030	Alpha & beta, save & load
sub-instructions		
BETAL	111,020	
SYSRET	062,030	System return
SYSMOV	rp,065	System save area move
SYSMOV BC	062,065	
SYSMOV DE	174,065	
SYSMOV HL	176,065	
SYSMOV XA	022,065	
DMPIN(141)		
DMPSIN(143)		
DMPOUT(145)		
DMPSOUT(147)		
BLKOUT(173)		
BLKIN(177)		
	1800 only	
UBOUT	0145	Output to device using Strobe 1
UBIN	0111 0145	Input from device using Strobe 1
UBOUT2	0062 0145	Output to device using Strobe 2
UBIN2	0113 0145	Input from device using Strobe 2

rs=source register
rd=destination register
(exp)=one-byte expression
(adr)=two-byte address
c=condition flag
nd,ns=register reference number
vvv=expression value
lsb=least significant byte of address
msb=most significant byte of address
rp=register pair
[rp]=register pair select code
[r]=register select code
(op)=arithmetic or logical operator: AD, AC, SU, SB, ND, OR, XR, CP
[pr]=register select code for paged instructions
(i)=expression for lsb of index address
(dsp)=expression for displacement of index instructions
[i]=lsb value of (i)
[srp]=special register pair select code for index instructions
BC=113
DE=115
HL=117

6600, 8600 Machine Instructions

Function	Operation Code	Description
BFLR(op)		Binary field left to right operations
BFLRAD	111 006	
BFLRAC	111 016	
BFLRSU	111 026	
BFLRSB	111 036	
BFLRND	111 046	
BFLRXR	111 056	
BFLROR	111 066	
D(op)M(rp)		Double memory to register operations
DADM(rp)	[rp] 013	
DACM(rp)	[rp] 310	
DSUM(rp)	[rp] 033	
DSBM(rp)	[rp] 330	
DNDM(rp)	[rp] 043	
DXRM(rp)	[rp] 053	
DORM(rp)	[rp] 063	
DCPM(rp)	[rp] 073	
DPLR(rp),loc		Double paged load reversed
DPLR BC,loc	062 114 LSP	
DPLR DE,loc	174 134 LSP	
DPLR HL,loc	176 154 LSP	
DPSR(rp),loc		Double paged store reversed
DPSR BC, loc	062 116 LSP	
DPSR DE, loc	174 136 LSP	
DPSR HL, loc	176 156 LSP	
STLO(r)		Sector table load starting at offset
STLOA	022 077	
STLOB	111 077	
STLOC	062 077	
STLOD	113 077	
STLOE	174 077	
INFO	111010	System information
D(op)P(rp),loc	Double paged to register operations	
DADP(rp),loc	[rp+1] 013 LOCLSB	
DACP(rp),loc	[rp+1] 310 LOCLSB	
DSUP(rp),loc	[rp+1] 033 LOCLSB	
DSBP(rp),loc	[rp+1] 330 LOCLSB	
DNDP(rp),loc	[rp+1] 043 LOCLSB	
DXRP(rp),loc	[rp+1] 053 LOCLSB	
DORP(rp),loc	[rp+1] 063 LOCLSB	
DCPP(rp),loc	[rp+1] 073 LOCLSB	
D(op)I(rp),data		Double immediate to register operations
DADI(rp),data	[rp]110 LSB MSB	

Function	Operation Code	Description
DACI(rp),data	[rp]311 LSB MSB	
DSUI(rp),data	[rp]130 LSB MSB	
DSBI(rp),data	[rp]331 LSB MSB	
DNDI(rp),data	[rp]140 LSB MSB	
DXRI(rp),data	[rp]150 LSB MSB	
DORI(rp),data	[rp]160 LSB MSB	
DCPI(rp),data	[rp]170 LSB MSB	
DM(op)(rp)		Double register to memory operations
DMAD(rp)	[rp+1] 110	
DMAC(rp)	[rp+1] 311	
DMSU(rp)	[rp+1] 130	
DMSB(rp)	[rp+1] 331	
DMND(rp)	[rp+1] 140	
DMXR(rp)	[rp+1] 150	
DMOR(rp)	[rp+1] 160	
P(op)(r),loc		Single paged to register operations
PAD(r),loc	[r] 106 LOCLSB	
PAC(r),loc	[r] 112 LOCLSB	
PSU(r),loc	[r] 122 LOCLSB	
PSB(r),loc	[r] 132 LOCLSB	
PND(r),loc	[r] 142 LOCLSB	
PXR(r),loc	[r] 152 LOCLSB	
POR(r),loc	[r] 162 LOCLSB	
PCP(r),loc	[r] 172 LOCLSB	
COMP(rp)		2's complement a register pair
COMP BC	062 011	
COMP DE	174 011	
COMP HL	176 011	
COMPS(rp)		2's complement a register pair
COMPS BC	113 011	
COMPS DE	115 011	
COMPS HL	117 011	
IMULT	111 011	Integer multiply: HLDE=HL*BC
IDIV	062 031	Integer divide: DE/BC=>Q(DE),R(HL)
DIDIV	111 031	Double integer divide: HLDE/BC=>Q(DE),R(HL)
LLDEL	111 051	Doubly linked list delete
LLINS	062 051	Doubly linked list insert

8800 Machine Instructions

Function	Operation Code	Description
L(r)	0d6(vvv)	Load immediate
L(rd)M	3d7	Load
L(rd)(rs)	3ds	Load
LM(rs)	37s	Load
L(rd)M(rp)		Load register from memory using BC, DE, or XA for address
LM(rs)(rp)	rp37s	Load register from memory using BC, DE, or XA for address
PL		Paged load
PL A,(loc)	105 LSP	
PL B,(loc)	114 LSP	
PL C,(loc)	124 LSP	
PL D,(loc)	134 LSP	
PL E,(loc)	144 LSP	
PL H,(loc)	154 LSP	
PL L,(loc)	164 LSP	
PS		Paged store
PS A,(loc)	107 LSP	
PS B,(loc)	116 LSP	
PS C,(loc)	126 LSP	
PS D,(loc)	136 LSP	
PS E,(loc)	146 LSP	
PS H,(loc)	156 LSP	
PS L,(loc)	166 LSP	
DL		Double load
DL DE,HL	047	
DL BC,HL	111 047	
DL BC,BC	062 047	
DL BC,DE	113 047	
DL DE,BC	174 047	
DL DE,DE	115 047	
DL HL,BC	176 047	
DL HL,DE	117 047	
DL HL,HL	057	
DS		Double store
DS DE,HL	027	
DS BC,HL	111 027	
DS BC,DE	113 027	
DS DE,BC	174 027	
DS HL,BC	176 027	
DS HL,DE	117 027	
DPL		Double paged load
DPL BC,(loc)	111 124 LSP	
DPL DE,(loc)	113 144 LSP	
DPL HL,(loc)	115 164 LSP	

Function	Operation Code	Description
DPS BC,(loc)	111 126 LSP	Double paged store
DPS DE,(loc)	113 146 LSP	
DPS HL,(loc)	115 166 LSP	
DPLR(rp),loc		Double paged load reversed
DPLR BC,loc	Ø62 114 LSP	
DPLR DE,loc	174 134 LSP	
DPLR HL,loc	176 154 LSP	
DPSR(rp),loc		Double paged store reversed
DPSR BC,loc	Ø62 116 LSP	
DPSR DE,loc	174 136 LSP	
DPSR HL,loc	176 156 LSP	
REGS	Ø55	Register store
REGL	111 Ø55	Register load
POP	Ø6Ø	Pop
PUSH	Ø7Ø	Push
PUSH(rp)	rp Ø7Ø	Push using BC, DE, or XA
PUSH loc	Ø51 (adr)	Push immediate
POP(rp)	rp Ø6Ø	Push using BC, DE, or XA
STKS	Ø65	Stack store
STKL	111 Ø65	Stack load
AD data	Ø04(vvv)	Add immediate
AD(rs),ADM	20s,207	Add
AC data	Ø14	Add with carry immediate
AC(rs),ACM	21s,217	Add
SU data	Ø24	Subtract immediate
SU(rs),SUM	22s,227	Subtract
SB data	Ø34	Subtract with borrow immediate
SB(rs),SBM	23s,237	Subtract with borrow
ND data	Ø44(vvv)	And immediate
ND(rs),NDM	24s,247	And
OR data	Ø64(vvv)	Or immediate
OR(rs),ORM	26s,267	Or
XR data	Ø54(vvv)	Exclusive or immediate
XR(rs),XRM	25s,257	Exclusive or
CP data	Ø74(vvv)	Compare immediate
CP(rs),CPM	27s,277	Compare
SRC	Ø02	Shift right circular
SRE	Ø32	Shift right extended
(op)(rs)(r)	r2ps	ADAB adds A to B
(op)M(r)	rØp>	ADM6 adds (HL) to C
(op)(r)(vvv)	rØp4	SVC 2Ø subtracts 2Ø from C
SRC(r)	rØ12	SRCB shifts B to right
SLC(r)	rØ02	SLCD shifts D to left
SRE(r)		SRED shifts D to right

Function	Operation Code	Description
P(op)(r),loc		Single paged to register operation
PAD(r),loc	[r]106 LOCLSB	
PAC(r),loc	[r]112 LOCLSB	
PSU(r),loc	[r]122 LOCLSB	
PSB(r),loc	[r]132 LOCLSB	
PND(r),loc	[r]142 LOCLSB	
PXR(r),loc	[r]152 LOCLSB	
POR(r),loc	[r]162 LOCLSB	
PCP(r),loc	[r]172 LOCLSB	
INCP		Increment register pair
INCP HL	Ø15	
INCP HL,2	117 Ø15	
INCP HL,A	Ø17	
INCP BC	Ø62 Ø15	
INCP BC,2	113 Ø15	
INCP BC,A	Ø62 Ø17	
INCP DE	174 Ø15	
INCP DE,2	115 Ø15	
INCP DE,A	174 Ø17	
INCP XA	Ø22 Ø15	
INCP XA,2	111 Ø15	
INCP XA,A	Ø22 Ø17	
DECP		Decrement register pair
DECP HL	Ø35	
DECP HL,2	117 Ø35	
DECP HL,A	Ø37	
DECP BC	Ø62 Ø35	
DECP BC,2	113 Ø35	
DECP BC,A	Ø62 Ø37	
DECP DE	174 Ø35	
DECP DE,2	115 Ø35	
DECP DE,A	174 Ø37	
DECP XA	Ø22 Ø35	
DECP XA,2	111 Ø35	
DECP XA,A	Ø22 Ø37	
D(op)M(rp)		Double memory to register operations
DADM(rp)	[rp] Ø13	
DACM(rp)	[rp] 31Ø	
DSUM(rp)	[rp] Ø33	
DSBM(rp)	[rp] 33Ø	
DNDM(rp)	[rp] Ø43	
DXRM(rp)	[rp] Ø53	
DORM(rp)	[rp] Ø63	
DCPM(rp)	[rp] Ø73	
D(op)P(rp),loc		Double paged to register operations
DADP(rp),loc	[rp+1] Ø13 LOCLSB	
DACP(rp),loc	[rp+1] 31Ø LOCLSB	

Function	Operation Code	Description
DSUP(rp),loc	[rp+1] Ø33	LOCLSB
DSBP(rp),loc	[rp+1] 33Ø	LOCLSB
DNDP(rp),loc	[rp+1] Ø43	LOCLSB
DXRP(rp),loc	[rp+1] Ø53	LOCLSB
DORP(rp),loc	[rp+1] Ø63	LOCLSB
DCPP(rp),loc	[rp+1] Ø73	LOCLSB

Assembler Directives--SNAP

(L)	EQU	(e)	Set value of label to (e)
	SET	(e)	Set and use ABSOLUTE PAB (ASSEMBLER 4: set LC)
	SKIP	(e)	Increment AC and LC by (e)
	TP		Tabulate AC and LC to next page boundary
	TM	(e)	Tabulate page if less than (e) bytes in present page
	DC	(e)	Generate 1-byte values for expressions (1 byte per character for string expressions)
	DA	(e)	Generate 2-byte values for expressions
	RPT	(e)	Repeat next line of code (e) times
	END	(e)	End assembly pass; (e) is program transfer address
	LIST	(e)	Set assembly listing control flags (L, F, G, I, M)
	.		Comment line (. in first column)
	+		Form feed, then print comment line
	*		Form feed if within 2" of end of page (comment line)
	INC	(e)	Include source file named by (e)
	LOC	(e),exp	Set LC to (e) and turn L flag on (exp not normally necessary)
	LOC	*,exp	Set LC to AC and turn L flag off (exp not normally necessary)
(L)	ORG	(e),flg	Set first and current word address of a new PAB named (L) (flags are T, P, C)
	USE	(e)	Use PAB (e), set AC to current word address of PAB
	USE	*	Revert to use of last PAB used
	ERR		Produce a P error
	IFnn	(e)	Turn assembly off, if condition "nn" is not met. Condition test compares first field of expression to second field. If second field not given, assume Ø Conditions: EQ - equal GT - greater than LT - less than NE - not equal GE - greater than or equal LE - less than or equal Z - field 1 zero NZ - field 1 not zero C - field 1 zero (clear) S - field 1 not zero (set) STR - field 1 begins with * NSTR - no * in field 1
	XIF		Turn assembly back on if it has been turned off
	TITLE		Page effect and print following line as title
	MLIB	(e)	Include macro library (e)
	MACRO		Macro definition follows
	MEND		End of macro definition
	ALIGN	(e)	Increment AC and LC to next memory location that is a multiple of (e) [(e)=2^n]
(L)	PROG		Label is name for following program module
	SNAPOPT		4, 2, 6, X, R
	TESTnnexp,(exp)		Pass 2 relation test

AC=Address counter
LC=Location counter
(L)=Label required
(e)=expression (allowed or required)

Labels--SNAP/3

Labels consist only of alphanumeric characters and $. A label must begin with an alpha character. Special terminating characters (not part of the label) indicate special qualities for the label.

Label Length (characters)	Over-length label action	Terminating Characters Character	Characteristic Assigned
8	uses first 7 and last 1 characters	*	program entry point
		=	redefinition
			external definition

Expressions

Numeric expressions use 16-bit two's complement values. If the instruction requires only one byte, the msb of the expression is discarded.

Expression evaluation is strictly left to right, all operators having equal precedence. SNAP/3 allows the use of parentheses to modify order of evaluation per normal algebraic convention.

Binary Operators		Unary Operators	
+	add	<	shift left number of places
-	subtract		indicated by next value
*	multiply	>	shift right number of places
/	integer division		indicated by next value
.AND.	logical AND	-	negation
.OR.	logical OR	*	set star flag
.XOR.	logical exclusive-OR		
.MOD.	remainder from division		

Strings can be included in all expressions. A string is delimited by apostrophes. The value of a character in a string is the ASCII value for the character with the parity bit (bit 7) always zero.

Note: Only the DC directive allows strings more than one character long. For this case, one byte of code is generated for each character.

Assembler Pseudo-instructions

Instruction	Expansion		Code
HL (e)	LL	lsb	Ø66 lsb Ø56 msb
	LH	msb	
DE (e)	LE	lsb	Ø46 lsb Ø36 msb
	LD	msb	
BC (e)	LC	lsb	Ø36 lsb Ø26 msb
	LB	msb	
XA (e)			
MSr_s (e)	LL	lsb	Ø66 lsb 37n_s
	LMr_s		
MSr_s *(e)	LL	lsb	Ø66 lsb Ø56 msb 37n_s
	LH	msb	
	LMr_s		
MLr_d (e)	LL	lsb	Ø66 lsb 3n_d7
	Lr_dM		
MLr_d *(e)	LL	lsb	Ø66 lsb Ø56 msb 3n_d7
	LH	msb	
	Lr_dM		
SRN (e)	RPT	(e)	Ø12 Ø12 ((e) times)
	SRC		
SLN (e)	RPT	(e)	ØØ2 ØØ2 ((e) times)
	SLC		
CCLr	ADnn		(r)2Ør

r_s=source register
r_d=destination register
(e)=expression
lsb=low-order byte of expression value
msb=high-order byte of expression value
n=register reference number

Assembly Error Flags--SNAP/3

D	Different definition of labels (pass 1 only, all but first occurence ignored on pass 2)
I	Instruction mnemonic undefined
E	Expression or label error (unrecognizable character)
U	Undefined label (value of zero assigned)
F	File error, inclusion limits exceeded or END found in included file
P	Programmer produced, ERR instruction encountered
>	Indicates external reference - not an error condition
O	Overflow on page sensitive PAB

MACROS

Macro Definition (Prototype)

```
     MACRO
  [label] name [symbol[(default)]][,symbol[(default)]]...
     .
     . code
     .
     MEND
```

Macro Call

```
     [label] name [expression][,expression]...
```

The replacement of symbols by expressions is position-dependent. If no expression is given for a symbol, the default replaces the symbol; if no default is given, the symbol disappears from the expanded code.

Macro names follow the same syntax rules as labels.

[] above encloses optional fields.

Macro Directives
MIFnn Identical to IFnn directive, for use only in macro definitions. (MIFnn compares strings, rather than numeric values.)
MXIF Identical to XIF directive, for use only in macro definitions.

Assembler Execution

SNAP/3 source[,object][,ept][,print][,include][;<option characters>]

A	Causes an absolute output file to be produced, instead of a relocatable file
D	Causes a source and object code listing to be displayed on the CRT; may be specified in addition to the L option.
F, G, I, M	Turns on corresponding listing control flags.
L	Produces a source and object code listing, which will appear on the local printer if neither the P, Q, nor S option appears.
P	Causes the L or X option listing to be to a print file.
Q	Same as P option, but specifies that the listing should be appended or queued after any information already in the print file.
S	Causes the L or X option listing to be to the local printer.
T	Forces a two-pass assembly and must be specified if the resulting relocatable output file is to be loaded by the DOS relocatable loader (DOS function 15).
X	Produces a cross-reference, map listing and may appear with or without the L option.
2, 6, U	Turns on the assembly options.

ROM Debug Entry Point Vectors

Ø167400	Memory Parity Failure Vector
Ø167406	Input Parity Failure Vector
Ø167414	Output Parity Failure Vector
Ø167422	Write Protect Violation Vector
Ø167430	Access Protect Violation Vector
Ø167436	Privileged Instruction Violation Vector
Ø167444	One Millisecond Clock Vector
Ø167452	User System Call Vector
Ø167460	Breakpoint Vector
Ø167466	Unassigned Instruction
Ø167474	Sector Table Parity Error

ROM Debug Display Format

AAAAAA	CURADR (The current address in octal)
X NNN	ASCII, 8-bit octal C (CURADR)
MMMMMM	16-bit (LSB, MSB) address formed at CURADR, CURADR + 1
nnnnnnn	Command entry position

6600 ROM Debug Command Summary

nnn	A	Address the (n) or current I/O device
nnn nnn	B	Set a breakpoint to the (n) or current address
nnn nnn	C	Call the (n) or current address
nnn nnn	D	Decrement the current address by (n) or 1
(nnn nnn)	E	Continue execution or replace top stack location with (n) and continue execution
nnn	F	Fetch next data byte from (n) or current device
nnn	G	Go to Data mode in (n) or current device on "E", "U", or "i" command
n	H	RIM buffer test for RIM number n, where n is from 1 to 6
(nnn) nnn	I	Increment the current address by (n) or current device on "E", "U" or "i" command
nnn nnn	J	Jump to given (n) or current address
12345	K	Set ASCII key-in mode
	L	Link to address pointed to by current address
(nnn) nnn	M	Modify the contents of the current address
nnnnnn	N	Set physical address to nnnnnn
nn	O	Select Origin entry table
[*](ENTER)		Set Origin addressing to entry value and display
[*](nnn)		Set Origin addressing to (n), enter in table and display
(nnn) nnn	P	Load Base register with (nnnnnn-0100000)>8
12345	Q	Load the sector table
	R	Switch ALPH/BETA mode and display
nn	S	Display the (nth) stack location item
12345	T	Start the primary 6600 memory test
nnn nnnn	U	Continue execution as in "E" command but in USER mode (Interrupts enabled.)
nnn	V	EX COM4 Device must be addressed for I/O commands
nn	W	EX WRITE status is displayed after command issue
nnn	X	EX COM1 (nnn) is the output byte
nnn	Y	EX COM2
nnn	Z	EX COM3
	?	Displays processor version, Micro-Code and Macro-Code revision levels
nnn	x	Display x register or modify to nnn
(nnn) nnn	a	A Modify register pair if
nnn	b	B argument exceeds eight bits
(nnn) nnn	c	C
nnn	d	D The LSB register specifies
(nnn) nnn	e	E the pair (i.e., L for H & L)
nnn	h	H
(nnn) nnn	l	L
nnn	f	Displays or updates the condition flags
nnn nnn	i	Same as 'E' above with interrupts enabled
nnn nnn	r	PUSH value (nnn nnn) onto stack
nn	s	POP stack (nn) times
nnn	p	Load base register direct with value (nnn)
12345	t	Alternate memory test
nnn	y	EX DATA (nnn) on output bus
nnn	z	EX STATUS (nnn) on output bus
nnnnnn	ENT	Set logical address to 'nnnnnn'
	CAN	Cancel entry line

BKSP	Backspace one on entry line
(nnn) nnn.	Modify and increment
nnn (nnn) ^	Modify and increment using the last non-null value

1800, 3800 ROM Debug Command Summary

A	Address given or last I/O device
B	Set breakpoint at given or current address
C	Call the given or current address (forces system mode)
D	Decrement the current address
E	Continue execution of program
F	Fetch next data byte from current I/O device
G	Go to data mode in the current device
*H	Hardware floppy diagnostic/3800 RIM buffer test
I	Increment current address
J	Jump to the given or current address
L	Link to address pointed to by current address
M	Modify the current address contents
P	Display base register or load W/value - 0100000
*Q	Load the sector table
R	Switch Alpha/Beta register mode
S	Display specified stack item
*T	Start memory test
U	Sets user mode and does an 'E' command
V	EX COM4 to last I/O device
W	EX WRITE to last I/O device
X	EX COM1 to last I/O device
Y	EX COM2 to last I/O device
Z	EX COM3 to last I/O device
a	A register display or set
b	B register display or set
c	C register display or set
d	D register display or set
e	E register display or set
f	Condition flags display
h	H register display or set
i	'E' command with EI/RET
j	Display test
l	L register display or set
p	Display base register or set with C
x	Register display
y	EX STATUS
z	EX DATA
?	Processor and Macro ROM type/version
.	'M' command followed by 'I' command
^	'.' using last value
#	Clear all break points

*Must be preceded by '12345' except 3800 RIM Buffer Test. Then the "H" must be preceded by the RIM number under test. (See device address assignments.)

1550 ROM Debug Command Summary

n A	Align diskette head
nnn nnn B	Set breakpoint at given orcurrent address
nnn nnn C	Call given or current address
nnn nnn D	Decrement current address
E	Continue execution
nnn F	Fill screen with given octal value
G	Display general communications channel status
H	SIO loopback test
nnn nnn I	Increment given or current address
nnn nnn J	Jump to given address
K	Keyin on top line of screen
nnn L	Link to given address orthe address pointed to by the current address
(nnn) nnn M	Modify contents of of the location pointed to by the current address
N	Set the current address to that of a two-byte area containing the # assigned to the interrupted task
12345O	Initiate a loopback test
P	Display printer channel status
12345 R	System reset
nnn S	Display specified stack item
12345 T	Start memory test
n V	Diskette verification
n X	Start continuous diskette controller buffer test
n Z	Diskette track Ø sensor alignment
(nnn) nnn .	'M' command followed by 'I'
<	Input a character from the channel last selected by a P/G or p/g command
nnn >	Output a character from the channel last selected by a P/G or p/g command to data port last selected
?	Identify firmware version number
nnn nnn<enter>	Change current address
#	Clear breakpoint
(nnn) nnn a	Display/modify contents of A register
nnn b	Display/modify contents of B register
(nnn) nnn c	Display/modify contents of C register
nnn d	Display/modify contents of D register
(nnn) nnn e	Display/modify contents of E register
nnn f	Display/modify contents of condition code F register
(nnn) nnn g	Write given command to general comm channel
nnn h	Display/modify contents of H register
(nnn) nnn l	Display/modify contents of L register
(nnn) nnn p	Write given command to printer channel
nnnnnn x	Display/modify contents of IX register
nnnnnn y	Display modify contents of IY register

8600 ROM Debug Command Summary

nnn nnn A	Set current I/O address to given or current address
nnn nnn B	Set breakpoint at given or current address
nnn nnn C	Call given or current address
nnn nnn D	Decrement current address by 1 or nnn nnn
nnn nnn E	Continue execution from nnn nnn
nnn nnn I	Increment current address by 1 or given
nnn nnn J	Jump to given or current address
nnn K	Set control register to nnn
(nnn) nnn M	Modify contents of current address location
nnn nnn N	Set current address to nnn nnn
nnn O	Set origin table pointer and origin mode
nnn nnn P	Load the base register with given
12345 Q	Load sector table selected by control register
R	Perform Alpha/Beta switch
nn S	Display stack entry nn
nnn T	Display sector table entry nnn
12345 T	Start processor and memory self-test
nnn Y	Modify or display the saved system status
Z	Display all registers and register pairs: FFF AAA BBB CCC DDD EEE HHH LLL XXX BBBCCC DDDEEE HHHLLL XXXAAA STKP
(nnn) nnn a	Modify/display A register
nnn b	Modify/display B register
(nnn) nnn c	Modify/display C register
nnn d	Modify/display D register
(nnn) nnn e	Modify/display E register
nn f	Modify/display condition code flags
(nnn) nnn h	Modify/display H register
nnn nnn i	Set addressing bias to given and select I/O space
nnn k	Modify/display saved control register to nnn
nnn l	Modify/display L register
nnn nnn o	Set addressing bias to nnn nnn and select memory space
nnn nnn p	Load saved base register with upper 8 bits of nnn nnn
nn r	Pop stack (nn) times
nnn nnn s	Push nnn nnn onto stack
12345 t	Go to diagnostic mini exec
nnn x	Modify/display X register
z	Display CP/RIM ID
nnn nnn ENTER	Set relative address in memory or I/O space
CANCEL	Cancel command input line
BACKSPACE	Backspace one space on input line
(nnn) nnn .	Modify memory and increment current address
(nnnnnn) ^	Same as '.' but save nnnnnn; if no nnnnnn, use the last nnnnnn saved
#	Clear all active debug set breakpoints
?	Display processor identification data

8800 ROM Debug Command Summary

nnnnnnØØ	Increment the current address by nnnnnn
nnnnnnØ1	Decrement the current address by nnnnnn
(nnn)nnnØ3	Modify the contents of the current address to nnn and of the next higher address to (nnn)
nnnnnnØ4	Display the program counter or modify PC to nnnnnn
(nnn)nnnØ5	Display the register pair XA or modify X to (nnn) and A to nnn and display
(nnn)nnnØ6	Display or modify BC
(nnn)nnnØ7	Display or modify DE
(nnn)nnnØ8	Display or modify HL
nnnØ9	Display or modify flags
nnn1Ø	EXSTATUS and output data byte nnn to addressed device
nnn11	EX DATA and output data byte nnn to addressed device
nnn12	EX WRITE and output data byte nnn to addressed device
nnn13	EX COM1 and output data byte nnn to addressed device
nnn14	EX COM2 and output data byte nnn to addressed device
nnn15	EX COM3 and output data byte nnn to addressed device
nnn16	EX COM4 and output data byte nnn to addressed device
nnn17	Address the given or current I/O device
nnn18	Fetches next data byte from current or given I/O device
19	Exit from DEBUG
nnnnnn21	Jump to address nnnnnn, default = current address
nnnnnn22	Modify the displayed (CMD29) scratch pad entry to nnnnnn
24	Switch from user to kernal sector table
25	Switch from kernal to user sector table
nnn26	Display the (nnn)th user stack entry, default = top stack entry
nnn27	Display the (nnn)th sector table entry, default = entry #Ø
nnn28	Modify the displayed sector table entry with nnn
nnnnnn29	Display the contents of scratch pad entry location nnnnnn
nnnnnn3Ø	Display the user stack pointer or modify to nnnnnn
nnnnnn31	Display the user sector table pointers from the saved state or modify to nnnnnn
nnnnnn32	Display the user stack bounds from the saved state or modify to nnnnnn
nnnnnn33	Display the user base register/instruction register from the saved state or modify to nnnnnn
34	Switch between the data and instruction sector tables for memory accesses
nnnnnnDSP	Display the contents of nnn and the next higher address, default = current address
(nnn)nnnMOD	Modify the current address by nnn and the next higher address by (nnn) and increment the address
BSP & ENTER	Cancels the entry line
BSP	Backspace by one character on the entry line

1130 ROM Debug Commands

(nnn) A	Address the I/O device specified by nnn and return with status
nnnnn C	Call memory address nnnnn returning to Debug
D	Decrement current address by 1
F	Fetch data byte from currently addressed I/O device
I	Increment current memory address by 1
nnnnn J	Jump to memory address nnnnn
L	Link to address pointed to by current address
(nnn)nnn M	Modify current address to given
O	Enter diskette test cycle
(nnn) S	Display contents of nnn level of hardware stack
T	Enter memory test cycle
nnn V	Execute EX COM4 to previously addressed I/O device, display status of A register
nnn W	Execute EX WRITE to previously addressed I/O device, display status of A register
nnn X	Execute EX COM1 to previously addressed I/O device, display status of A register. For diskettes: ØX to 3X - Select drive Ø to 3 4X - Clear parity error flag 5X - Read sector from diskette to buffer 6X - Write sector from buffer to diskette 7X - Write verify sector from buffer to diskette 1ØX - Restore selected drive n11X - Select buffer page n (Ø to 3)
nnn Y	Execute EX COM2 to previously addressed I/O device, display status of A register
nnn Z	Execute EX COM3 to previously addressed I/O device, display status of A register
a	Display A register
b	Display B register
c	Display C register
d	Display D register
e	Display E register
f	Display condition codes
h	Display H register
l	Display L register
(nnn) .	Modify current address of memory to nnn
(nnnnn) ENTER	Modify current address to nnnnn
CANCEL	Cancel entry line

1150/1170 ROM Debug Command Summary

nnn A	Address given or current I/O device
nnn nnn B	Set breakpoint at given or current address
nnn nnn C	Call given or current address
nnn nnn D	Decrement current address by 1 or nnn nnn
nnn nnn E	Continue execution
nnn F	Fetch next data byte from current or given I/O device
nnn G	Go to data mode in current or given I/O device when 'E' command is given
12345 H	Start 1150 system diagnostic test
nnn nnn I	Increment current address by 1 or nnn nnn
nnn nnn J	Jump to given current address
12345 K	Set ASCII keyin mode
L	Link to address pointed to by current address
(nnn) nnn M	Modify contents of current address location
nn O	Selects given into origin table
nnn nnn P	Load base register with 8-bit value
12345 Q	Load sector table
R	Switch Alpha/Beta mode register display
nn S	Display given stack item
12345 T	Start primary memory test
nnn nnn U	User mode execute with optional return to nnn nnn address
nnn V	EX COM4 Device must be address with a command
nnn W	EX WRITE Status is displayed
nnn X	EX COM1 after the command is issued
nnn Y	EX COM2 'nnn' is current output byte.
nnn Z	EX COM3
nnn x	Display/modify X register
(nnn) nnn a	Display/modify A register
nnn b	Display/modify B register
(nnn) nnn c	Display/modify C register
nnn d	Display/modify D register
(nnn) nnn e	Display/modify E register
nnn h	Display/modify H register
(nnn) nnn l	Display/modify L register
nnn f	Display/modify condition flag
nnn nnn i	Same as 'E' but with interrupts enabled
nnn nnn h	PUSH given value onto stack
nn s	POP stack (nn) times
nnn p	Load base register with given value
12345 t	Start pseudo-random memory test
nnn y	EX DATA with given on output bus
nnn z	EX STATUS with given on output bus
nnn nnn ENTER	Set current address to given
CANCEL	Cancel entry line
BACKSPACE	Backspace on entry line
(nnn) nnn .	Modify contents and increment current address
(nnn) nnn ^	Modify contents and increment current address
#	Clear all breakpoints

CHARACTER TRANSMISSION AND TRANSLATION TABLE

Dec	Octal	Hex	ASCII	EBCDIC	IBM BCD	Honeywell BCD	EBCDIC Card Code	Binary
0	000	00	NUL	NUL		0	12-0-1-8-9	00 000 000
1	001	01	SOH	SOH	1	1	12-1-9	00 000 001
2	002	02	STX	STX	2	2	12-2-9	00 000 010
3	003	03	ETX	ETX	3	3	12-3-9	00 000 011
4	004	04	EOT	PF	4	3	12-4-9	00 000 100
5	005	05	ENQ	HT	5	5	12-5-9	00 000 101
6	006	06	ACK	LC	6	6	12-6-9	00 000 110
7	007	07	BEL	DEL	7	7	12-7-9	00 000 111
8	010	08	BS		8	8	12-8-9	00 001 000
9	011	09	HT	RLF	9	9	12-1-8-9	00 001 001
10	012	0A	LF	SMM	0	1	12-2-8-9	00 001 010
11	013	0B	VT	VT	=	=	12-3-8-9	00 001 011
12	014	0C	FF	FF	/	:	12-4-8-9	00 001 100
13	015	0D	CR	CR	:		12-5-8-9	00 001 101
14	016	0E	SO	SO	>	>	12-6-8-9	00 001 110
15	017	0F	SI	SI	@	&	12-7-8-9	00 001 111
16	020	10	DLE	DLE	Space	+	12-11-1-8-9	00 010 000
17	021	11	DC1	DC1	/	A	11-1-9	00 010 001
18	022	12	DC2	DC2	S	B	11-2-9	00 010 010
19	023	13	DC3	TM/DC3	T	C	11-3-9	00 010 011
20	024	14	DC4	RES	U	D	11-4-9	00 010 100
21	025	15	NAK	NL	V	E	11-5-9	00 010 101
22	026	16	SYN	BS	W	F	11-6-9	00 010 110
23	027	17	ETB	IL	X	G	11-7-9	00 010 111
24	030	18	CAN	CAN	Y	H	11-8-9	00 011 000
25	031	19	EM	Em	Z	I	11-1-8-9	00 011 001
26	032	1A	SUB	CC	=	;	11-2-8-9	00 011 010
27	033	1B	ESC	CU1	.	.	11-3-8-9	00 011 011
28	034	1C	FS	IFS	(	)	11-4-8-9	00 011 100
29	035	1D	GS	IGS		%	11-5-8-9	00 011 101
30	036	1E	RS	IRS	¬	dirty loz.	11-6-8-9	00 011 110
31	037	1F	US	IUS	"	?	11-7-8-9	00 011 111
32	040	20	SPACE	DS	-	-	11-0-1-8-9	00 100 000
33	041	21	!	SOS	J	J	0-1-9	00 100 001
34	042	22	"	FS	K	K	0-2-9	00 100 010
35	043	23	#		L	L	0-3-9	00 100 011
36	044	24	$	BYP	M	M	0-4-9	00 100 100
37	045	25	%	LF	N	N	0-5-9	00 100 101
38	046	26	&	ETB	O	O	0-6-9	00 100 110
39	047	27	'	ESC	P	P	0-7-9	00 100 111

Dec	Octal	HEX	ASCII	EBCDIC	IBM BCD	Honeywell BCD	EBCDIC Card Code	Binary
40	050	28	(		Q	Q	0-8-9	00 101 000
41	051	29	)		R	R	0-1-8-9	00 101 001
42	052	2A	*	SM	!	#	0-2-8-9	00 101 010
43	053	2B	+	CU2	$	$	0-3-8-9	00 101 011
44	054	2C	,		*	*	0-4-8-9	00 101 100
45	055	2D	-	ENQ	]	"	0-5-8-9	00 101 101
46	056	2E	.	ACK	;	=	0-6-8-9	00 101 110
47	057	2F	/	BEL	^	!	0-7-8-9	00 101 111
48	060	30	0		+	<	12-11-0-1-8-9	00 110 000
49	061	31	1		A	/	1-9	00 110 001
50	062	32	2	SYN	B	S	2-9	00 110 010
51	063	33	3		C	T	3-9	00 110 011
52	064	34	4	PN	D	U	4-9	00 110 100
53	065	35	5	RS	E	V	5-9	00 110 101
54	066	36	6	UC	F	W	6-9	00 110 110
55	067	37	7	EOT	G	X	7-9	00 110 111
56	070	38	8		H	Y	8-9	00 111 000
57	071	39	9		I	Z	1-8-9	00 111 001
58	072	3A	:		?	@	2-8-9	00 111 010
59	073	3B	;	CU3	.	.	3-8-9	00 111 011
60	074	3C	<	DC4	)	CR	4-8-9	00 111 100
61	075	3D	=	NAK	[	clean loz.	5-8-9	00 111 101
62	076	3E	>		<	¢	6-8-9	00 111 110
63	077	3F	?	SUB	#		7-8-9	00 111 111
64	100	40	@	SPACE			NO PUNCHES	01 000 000
65	101	41	A				12-0-1-9	01 000 001
66	102	42	B				12-0-2-9	01 000 010
67	103	43	C				12-0-3-9	01 000 011
68	104	44	D				12-0-4-9	01 000 100
69	105	45	E				12-0-5-9	01 000 101
70	106	46	F				12-0-6-9	01 000 110
71	107	47	G				12-0-7-9	01 000 111
72	110	48	H				12-0-8-9	01 001 000
73	111	49	I				12-1-8	01 001 001
74	112	4A	J	¢			12-2-8	01 001 010
75	113	4B	K	.			12-3-8	01 001 011
76	114	4C	L	<			12-4-8	01 001 100
77	115	4D	M	(			12-5-8	01 001 101
78	116	4E	N	+			12-6-8	01 001 110
79	117	4F	O	|			12-7-8	01 001 111

Dec	Octal	HEX	ASCII	EBCDIC	IBM BCD	Honeywell BCD	EBCDIC Card Code	Binary
80	120	50	P	&			12	01 010 000
81	121	51	Q				12-11-1-9	01 010 001
82	122	52	R				12-11-2-9	01 010 010
83	123	53	S				12-11-3-9	01 010 011
84	124	54	T				12-11-4-9	01 010 100
85	125	55	U				12-11-5-9	01 010 101
86	126	56	V				12-11-6-9	01 010 110
87	127	57	W				12-11-7-9	01 010 111
88	130	58	X				12-11-8-9	01 011 000
89	131	59	Y				11-1-8	01 011 001
90	132	5A	Z	!			11-2-8	01 011 010
91	133	5B	[	$			11-3-8	01 011 011
92	134	5C	◇	*			11-4-8	01 011 100
93	135	5D	]	)			11-5-8	01 011 101
94	136	5E	^	;			11-6-8	01 011 110
95	137	5F	_	¬			11-7-8	01 011 111
96	140	60	\	-			11	01 100 000
97	141	61	a	/			0-1	01 100 001
98	142	62	b				11-0-2-9	01 100 010
99	143	63	c				11-0-3-9	01 100 011
100	144	64	d				11-0-4-9	01 100 100
101	145	65	e				11-0-5-9	01 100 101
102	146	66	f				11-0-6-9	01 100 110
103	147	67	g				11-0-7-9	01 100 111
104	150	68	h				11-0-8-9	01 101 000
105	151	69	i	¦			0-1-8	01 101 001
106	152	6A	j	¦			12-11	01 101 010
107	153	6B	k	,			0-3-8	01 101 011
108	154	6C	l	%			0-4-8	01 101 100
109	155	6D	m	_			0-5-8	01 101 101
110	156	6E	n	>			0-6-8	01 101 110
111	157	6F	o	?			0-7-8	01 101 111
112	160	70	p				12-11-0	01 110 000
113	161	71	q				12-11-0-1-9	01 110 001
114	162	72	r				12-11-0-2-9	01 110 010
115	163	73	s				12-11-0-3-9	01 110 011
116	164	74	t				12-11-0-4-9	01 110 100
117	165	75	u				12-11-0-5-9	01 110 101
118	166	76	v				12-11-0-6-9	01 110 110
119	167	77	w				12-11-0-7-9	01 110 111

Dec	Octal	HEX	ASCII	EBCDIC	IBM BCD	Honeywell BCD	EBCDIC Card Code	Binary
120	170	78	x				12-11-0-8-9	01 111 000
121	171	79	y	/			1-8	01 111 001
122	172	7A	z	:			2-8	01 111 010
123	173	7B	{	#			3-8	01 111 011
124	174	7C	:	@			4-8	01 111 100
125	175	7D	}	'			5-8	01 111 101
126	176	7E	~	=			6-8	01 111 110
127	177	7F	DEL	"			7-8	01 111 111
128	200	80					12-0-1-8	10 000 000
129	201	81		a			12-0-1	10 000 001
130	202	82		b			12-0-2	10 000 010
131	203	83		c			12-0-3	10 000 011
132	204	84		d			12-0-4	10 000 100
133	205	85		e			12-0-5	10 000 101
134	206	86		f			12-0-6	10 000 110
135	207	87		g			12-0-7	10 000 111
136	210	88		h			12-0-8	10 001 000
137	211	89		i			12-0-9	10 001 001
138	212	8A					12-0-2-8	10 001 010
139	213	8B					12-0-3-8	10 001 011
140	214	8C					12-0-4-8	10 001 100
141	215	8D					12-0-5-8	10 001 101
142	216	8E					12-0-6-8	10 001 110
143	217	8F					12-0-7-8	10 001 111
144	220	90					12-11-1-8	10 010 000
145	221	91		j			12-11-1	10 010 001
146	222	92		k			12-11-2	10 010 010
147	223	93		l			12-11-3	10 010 011
148	224	94		m			12-11-4	10 010 100
149	225	95		n			12-11-5	10 010 101
150	226	96		o			12-11-6	10 010 110
151	227	97		p			12-11-7	10 010 111
152	230	98		q			12-11-8	10 011 000
153	231	99		r			12-11-9	10 011 001
154	232	9A					12-11-2-8	10 011 010
155	233	9B					12-11-3-8	10 011 011
156	234	9C					12-11-4-8	10 011 100
157	235	9D					12-11-5-8	10 011 101
158	236	9E					12-11-6-8	10 011 110
159	237	9F					12-11-7-8	10 011 111

Dec	Octal	HEX	ASCII	EBCDIC	IBM BCD	Honeywell BCD	EBCDIC Card Code	Binary
160	240	A0					11-0-1-8	10 100 000
161	241	A1		~			11-0-1	10 100 001
162	242	A2		s			11-0-2	10 100 010
163	243	A3		t			11-0-3	10 100 011
164	244	A4		u			11-0-4	10 100 100
165	245	A5		v			11-0-5	10 100 101
166	246	A6		w			11-0-6	10 100 110
167	247	A7		x			11-0-7	10 100 111
168	250	A8		y			11-0-8	10 101 000
169	251	A9		z			11-0-9	10 101 001
170	252	AA					11-0-2-8	10 101 010
171	253	AB					11-0-3-8	10 101 011
172	254	AC					11-0-4-8	10 101 100
173	255	AD					11-0-5-8	10 101 101
174	256	AE					11-0-6-8	10 101 110
175	257	AF					11-0-7-8	10 101 111
176	260	B0					12-11-0-1-8	10 110 000
177	261	B1					12-11-0-1	10 110 001
178	262	B2					12-11-0-2	10 110 010
179	263	B3					12-11-0-3	10 110 011
180	264	B4					12-11-0-4	10 110 100
181	265	B5					12-11-0-5	10 110 101
182	266	B6					12-11-0-6	10 110 110
183	267	B7					12-11-0-7	10 110 111
184	270	B8					12-11-0-8	10 111 000
185	271	B9					12-11-0-9	10 111 001
186	272	BA					12-11-0-2-8	10 111 010
187	273	BB					12-11-0-3-8	10 111 011
188	274	BC					12-11-0-4-8	10 111 100
189	275	BD					12-11-0-5-8	10 111 101
190	276	BE					12-11-0-6-8	10 111 110
191	277	BF					12-11-0-7-8	10 111 111
192	300	C0					12-0	11 000 000
193	301	C1		A			12-1	11 000 001
194	302	C2		B			12-2	11 000 010
195	303	C3		C			12-3	11 000 011
196	304	C4		D			12-4	11 000 100
197	305	C5		E			12-5	11 000 101
198	306	C6		F			12-6	11 000 110
199	307	C7		G			12-7	11 000 111

Dec	Octal	HEX	ASCII	EBCDIC	IBM BCD	Honeywell BCD	EBCDIC Card Code	Binary
200	310	C8		H			12-8	11 001 000
201	311	C9		I			12-9	11 001 001
202	312	CA					12-0-2-8-9	11 001 010
203	313	CB					12-0-3-8-9	11 001 011
204	314	CC					12-0-4-8-9	11 001 100
205	315	CD					12-0-5-8-9	11 001 101
206	316	CE					12-0-6-8-9	11 001 110
207	317	CF					12-0-7-8-9	11 001 111
208	320	D0					11-0	11 010 000
209	321	D1		J			11-1	11 010 001
210	322	D2		K			11-2	11 010 010
211	323	D3		L			11-3	11 010 011
212	324	D4		M			11-4	11 010 100
213	325	D5		N			11-5	11 010 101
214	326	D6		O			11-6	11 010 110
215	327	D7		P			11-7	11 010 111
216	330	D8		Q			11-8	11 011 000
217	331	D9		R			11-9	11 011 001
218	332	DA					12-11-2-8-9	11 011 010
219	333	DB					12-11-3-8-9	11 011 011
220	334	DC					12-11-4-8-9	11 011 100
221	335	DD					12-11-5-8-9	11 011 101
222	336	DE					12-11-6-8-9	11 011 110
223	337	DF					12-11-7-8-9	11 011 111
224	340	E0		◇			0-2-8	11 100 000
225	341	E1					11-0-1-9	11 100 001
226	342	E2		S			0-2	11 100 010
227	343	E3		T			0-3	11 100 011
228	344	E4		U			0-4	11 100 100
229	345	E5		V			0-5	11 100 101
230	346	E6		W			0-6	11 100 110
231	347	E7		X			0-7	11 100 111
232	350	E8		Y			0-8	11 101 000
233	351	E9		Z			0-9	11 101 001
234	352	EA					11-0-2-8-9	11 101 010
235	353	EB					11-0-3-8-9	11 101 011
236	354	EC					11-0-4-8-9	11 101 100
237	355	ED					11-0-5-8-9	11 101 101
238	356	EE					11-0-6-8-9	11 101 110
239	357	EF					11-0-7-8-9	11 101 111

Dec	Octal	HEX	ASCII	EBCDIC	IBM BCD	Honeywell BCD	EBCDIC Card Code	Binary
240	360	F0		0			0	11 110 000
241	361	F1		1			1	11 110 001
242	362	F2		2			2	11 110 010
243	363	F3		3			3	11 110 011
244	364	F4		4			4	11 110 100
245	365	F5		5			5	11 110 101
246	366	F6		6			6	11 110 110
247	367	F7		7			7	11 110 111
248	370	F8		8			8	11 111 000
249	371	F9		9			9	11 111 001
250	372	FA					12-11-0-2-8-9	11 111 010
251	373	FB					12-11-0-3-8-9	11 111 011
252	374	FC					12-11-0-4-8-9	11 111 100
253	375	FD					12-11-0-5-8-9	11 111 101
254	376	FE					12-11-0-6-8-9	11 111 110
255	377	FF					12-11-0-7-8-9	11 111 111